**Path finder Problem**

Chuanwei Wu

*Abstract* – In this project, I am asked to path finder, the path finder will be able to find the path which contains the most information, also known as the concentration of the methane, CH4. The concentration of target will be recorded and stored for the further analysis. The trajectory of the sensor is plotted to compare with the random input trajectory. The cumulative concentration information gathering change during the test is also plotted as result shown in the following part of the paper. The path designed in this project is user friendly as the sensor can catch the most methane leakage information. The path is assumed to be always starting from the upper left corner of the map, aka (0,5000).

1. **INTRODUCTION AND THEORY**

Methane, CH4, is twenty times more effective than carbon dioxide in trapping heat in the atmosphere over a 100-year period. Because of its global warming potential, recent efforts have focused on the prevention and detection of methane emissions. In 2010, the largest anthropogenic source of methane emissions in the United States was the natural gas and oil industry. Approximately 35% of the methane released by these systems occurs during the field exploration and production stages of the natural gas and oil extraction processes. In this stage, the release of methane is primarily attributed to fugitive emissions, which are uncontrolled and/or unintended gas releases. Determining the sources of fugitive methane emissions is integral in reducing the impact of the natural gas industry on the environment, and, as the distance between sources and populations decrease, reducing the risks to human health. The monitoring of oil fields, like the one illustrated in *figure 1*, by a mobile sensor is the motivation for this project.



Figure Illustrative example of oil fields and roadmap.

This project addresses the problem of detecting methane emissions from distributed natural gas and oil field production sites using a mobile sampling vehicle that can measure methane concentrations along public roadways over a desired region of interest (ROI). The path is asked to be determined based on the given road system illustrated in *figure 2*.

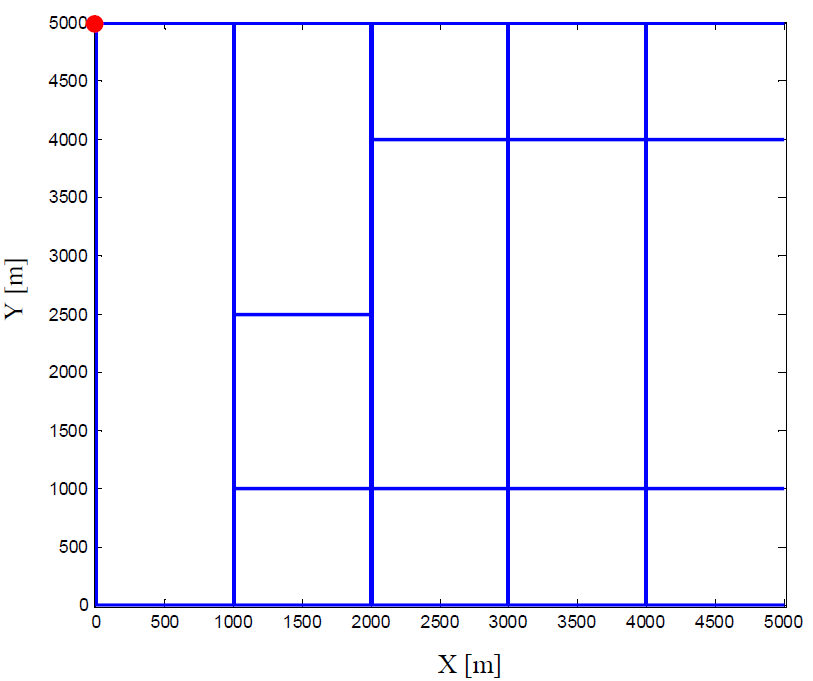


Figure Road map for ROI (roads are shown in blue solid lines), and initial vehicle

To identify leaking wells, the path designed is required to maximize the measurement value for the estimations of methane concentrations in *figure 3*. Let C denote the CH4 concentration in parts-per-million [ppm]. Because, in this case, the sensor is deployed to determine methane leaks, the instantaneous measurement benefit can be assumed equal to C, and, thus, it is a function of x and t. All the wells are assumed to leak at the same high rate, the concentration field in *figure 3* is determined using a forward Gaussian dispersion model. The instantaneous cost of the mobile sensor is assumed proportional to the linear speed, v, which could be a function of x, as per feedback control law, and is typically a function of time. If the concentration field can be assumed to remain constant over a period [t0, tf], the sensor objective function can be written as,

*Formula 1*

where wc is assumed to be 1, while wd is assumed to be 0. In other words, the power consumption is assumed to be negligible, all we care about the project is to gather the more information the possible. As the maximum speed is set to be 10 km/hr, the maximum distance travelled by the sensor is depended on the time travelled. In this project, the period is set to be 30 minutes, 45 minutes, 2 hours and 4 hours.



Figure Concentration field over the ROI in Fig. 2, assuming all wells leak at the same high rate.

For simplicity, it is assumed that the vehicle obeys the unicycle dynamics,

*Formula 2*

where v and are the control inputs, and x = [x y]T is the sensor position. The initial position of the vehicle, is illustrated by the red dot in *figure 2*, where x0 = 0 [m], and y0 = 5000 [m]. Assume a perfect measurement model by which the sensor measures the exact concentration of methane at every position, x, visited by the vehicle over time.

1. **Numerical Methods**

In the beginning of the code, the length of path is set to be velocity times the time divided by 50, which indicating how many blocks will the path covering. As the power consumption is set to be ignored, the path length will be directly related to the time allowed.

As all the information is known on the map, and the map is discrete system with 10,000 blocks, I treated the problem as a state machine problem. As each block, I just think which way is the most beneficial way to go based on the given information. Another thing to set in the map is to change all the non-path road information gathering to be 0. In this way, I can simply tell the system not to go to the 0 position in the original set-up map. After I visited a block, I will set that block to be 0 too, as it will avoid the system to be trapped in “local minimum”. One thing to be noticed is that the new set 0 is different from the original 0-value in the map, as the original 0-value block is the place that the sensor cannot visit, while the new-set 0 is the place the system prefer not to visit.

Another big change on the concentration map is that, on the path, there are some 0-valued block originally. To avoid confusion for the sensor, I changed those values to be 0.01. The result will not be affect a lot, while the performance of the algorithm will be much better.

The estimation at each block I made is based on the average concentration I will get on each direction. The mean function is used to make the calculation.

1. **Results**

The path is found for the period to be 30 minutes, 45 minutes, 2 hours and 4 hours. For the first concentration plot, the result is shown below,

|  |  |
| --- | --- |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\0.75_1.jpg  Figure The path plan for time of 30 minutes | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\0.5_1.jpg  Figure The path plan for time of 45 minutes |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2_1.jpg  Figure The path plan for time of 2 hours | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\4_1.jpg  Figure The path plan for time of 4 hours |

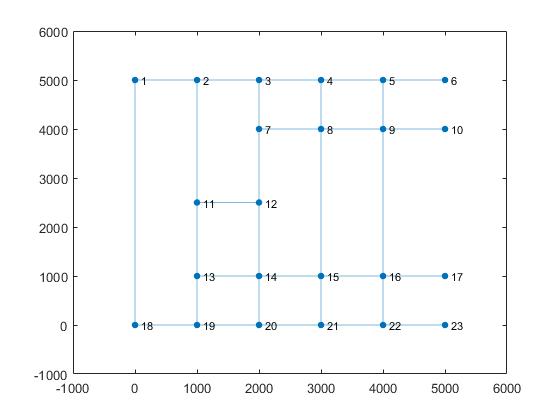


Figure The connectivity graph of the road map

For the 2 hours and 4 hours ones, the path is non-deterministic, as the same road is revisited for several times. For the 2-hour path, it goes like node 1 -> node 2-> node 13 -> node 14 -> biggest value between node 14 and node 20 -> node 7 -> node 8 -> node 4 -> biggest value between node 5 and node 6 -> node 5 -> biggest value between node 16 and node 22 -> node 16 -> biggest value between node 16 and node 17 -> biggest value between node 15 and node 16.

For the 4-hour path, it goes like node 1 -> node 2-> node 13 -> node 14 -> biggest value between node 14 and node 20 -> node 7 -> node 8 -> node 4 -> biggest value between node 5 and node 6 -> node 5 -> biggest value between node 16 and node 22 -> node 16 -> biggest value between node 16 and node 17 -> node 15 -> node 8-> node 7 -> node 14 -> node 13 -> node 17 -> node 13 -> node 19 -> node 18 -> node 1 (didn’t finish since the time is up) The connectivity graph is shown as *figure 8*,

The concentration information gathered along the path is plotted as *following*,

|  |  |
| --- | --- |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con1_0.5.jpg  Figure Concentration information gathered in 30 minutes on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon1_0.5.jpg  Figure Cumulative concentration in 30 minutes on all leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con1_0.75.jpg  Figure Concentration information gathered in 45 minutes on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon1_0.75.jpg  Figure Cumulative concentration in 45 minutes on all leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con1_2.jpg  Figure Concentration information gathered in 2 hours on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon1_2.jpg  Figure Cumulative concentration in 2 hours on all leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con1_4.jpg  Figure Concentration information gathered in 4 hours on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon1_4.jpg  Figure Cumulative concentration in 4 hours on all leakage well situation |

If the wells are assumed to be 10% leakage, then the concentration graph will be like following,



Figure Concentration field over the ROI in Fig. 2, assuming a random 10% of the wells leak at the same high rate, where the same color map in Fig. 3 expresses C in [ppm].

If the same paths are applied to the new situation, the concentration and cumulative concentration graph will be like following,

|  |  |
| --- | --- |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con2_0.5.jpg  Figure Concentration information gathered in 30 minutes on 10% leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon2_0.5.jpg  Figure Cumulative concentration in 30 minutes on 10% leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con2_0.75.jpg  Figure Concentration information gathered in 45 minutes on 10% leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon2_0.75.jpg  Figure Cumulative concentration in 45 minutes on 10% leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con2_2.jpg  Figure Concentration information gathered in 2 hours on 10% leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon2_2.jpg  Figure Cumulative concentration in 2 hours on 10% leakage well situation |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con2_4.jpg  Figure Concentration information gathered in 4 hours on 10% leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon2_4.jpg  Figure Cumulative concentration in 4 hours on 10% leakage well situation |

The random method is also used to find a path for the time of 45 minutes. Random is defined as at each cross-road, four random number will be generated and compared to each other. In this way, the system will decide which way to go. The result is shown below, we can tell the algorithm used in this project is way better than the random method. **(answer to the Q3)**

In our application and model, the final time will affect the information amount gathering. As the power consumption is neglected in this method, longer the time, the more length will be covered. In this way, more information will be gathered. In general, if enough time is allowed, all the information will be gathered. **(answer to the Q4)**

As all the information distribution is known, the only missing piece in this project is the weather condition. If the wind speed and wind direction is known, the algorithm can be improved to gather more information with lower cost. Since, sometime, the mobile sensor can stay in one location, waiting for the wind to bring the information towards it. **(answer to the Q5)**

|  |  |
| --- | --- |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\0.75_1.jpg  Figure The path plan for time of 45 minutes | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ran_path.jpg  Figure The path plan for time of 45 minutes by using random method |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\con1_0.75.jpg  Figure Concentration information gathered in 30 minutes on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ran_con.jpg  Figure Concentration information gathered in 30 minutes on all leakage well situation by using random method |
| C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cumcon1_0.75.jpg  Figure Cumulative concentration in 2 hours on all leakage well situation | C:\Users\Chuanwei Wu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ran_cum.jpg  Figure Cumulative concentration in 2 hours on all leakage well situation by using random method |

1. **Conclusions**

In this project, the path planer is designed. By using this planer, the mobile sensor can follow the path in the workspace and record the related information. For all the possible path, the path finder can catch the most information-rich path, which is indicating the robustness of the algorithm and the algorithm. The information is assumed to on the path of the road map, which may not be case in the real-world application. In the future edition of the path finder, the program should be able to take power consumption into consideration, , so it can find the more reasonable path. To simplify the project, noise is not existing in this project. In the real world, how to get rid of the noise is actually a large part of the sensor function. So it might be more considerate if the noise can be taken into consideration for the future edition of the path finder.

**Appendix**

mapSetup

workSpace=zeros(23);

workSpace(1,2)=1;

workSpace(1,18)=1;

workSpace(2,3)=1;

workSpace(2,11)=1;

workSpace(3,4)=1;

workSpace(3,7)=1;

workSpace(4,8)=1;

workSpace(4,5)=1;

workSpace(5,9)=1;

workSpace(5,6)=1;

workSpace(7,8)=1;

workSpace(7,12)=1;

workSpace(8,9)=1;

workSpace(8,15)=1;

workSpace(9,10)=1;

workSpace(9,16)=1;

workSpace(11,12)=1;

workSpace(11,13)=1;

workSpace(12,14)=1;

workSpace(13,14)=1;

workSpace(13,19)=1;

workSpace(14,15)=1;

workSpace(14,20)=1;

workSpace(15,16)=1;

workSpace(15,21)=1;

workSpace(16,17)=1;

workSpace(16,22)=1;

workSpace(18,19)=1;

workSpace(19,20)=1;

workSpace(20,21)=1;

workSpace(21,22)=1;

workSpace(22,23)=1;

count=0;

xPosition=[0 1000 2000 3000 4000 5000 2000 3000 4000 5000 1000 2000 1000 2000 3000 4000 5000 0 1000 2000 3000 4000 5000];

yPosition=[5000 5000 5000 5000 5000 5000 4000 4000 4000 4000 2500 2500 1000 1000 1000 1000 1000 0 0 0 0 0 0];

weight=zeros(32,1);

for i=1:23

for j=1:23

if workSpace(i,j)==1

count=count+1;

weight(count)=abs(xPosition(i)-xPosition(j)+yPosition(i)-yPosition(j));

end

end

end

workSpace=workSpace+workSpace';

figure

g=graph(workSpace);

h=plot(g);

xPosition=[0 1000 2000 3000 4000 5000 2000 3000 4000 5000 1000 2000 1000 2000 3000 4000 5000 0 1000 2000 3000 4000 5000];

yPosition=[5000 5000 5000 5000 5000 5000 4000 4000 4000 4000 2500 2500 1000 1000 1000 1000 1000 0 0 0 0 0 0];

h.XData=xPosition;

h.YData=yPosition;

g.Edges(:,2)=array2table(weight);

Main code

clc

clear

close all

mapSetup

vMax=10000;

time=0.75;

load('C\_all\_leaking.mat');

realPath=zeros(100);

C(C==0)=0.01;

for i=0:4

realPath(:,1+i\*20)=C(:,i\*20+1);

end

realPath(1,:)=C(1,:);

realPath(21,41:end)=C(21,41:end);

realPath(51,21:41)=C(51,21:41);

realPath(81,21:end)=C(81,21:end);

realPath(100,:)=C(100,:);

realPath1=zeros(102);

realPath1(2:end-1,2:end-1)=realPath;

realPath2=realPath1;

% mean(mean(realPath))

% sum(sum(realPath~=0))

% mean(mean(realPath(realPath~=0)))

position(1,:)=[2 2];

count=0;

figure;hold on

plot([100 100],[0 5000],'k','LineWidth',1)

plot([100 5100],[5000 5000],'k','LineWidth',1)

plot([100 5100],[50 50],'k','LineWidth',1)

plot([1100 1100],[0 5000],'k','LineWidth',1)

plot([2100 2100],[0 5000],'k','LineWidth',1)

plot([3100 3100],[0 5000],'k','LineWidth',1)

plot([4100 4100],[0 5000],'k','LineWidth',1)

plot([2100 5100],[4000 4000],'k','LineWidth',1)

plot([1100 5100],[1000 1000],'k','LineWidth',1)

plot([1100 2100],[2500 2500],'k','LineWidth',1)

xlim([0,5200])

ylim([0,5200])

for i=1:vMax\*time/50

if i <536

down=mean(realPath1(position(i,1)+1:end,position(i,2)));

right=mean(realPath1(position(i,1),position(i,2)+1:end));

up=mean(realPath1(1:position(i,1)-1,position(i,2)));

left=mean(realPath1(position(i,1),1:position(i,2)-1));

if max([down,right,up,left])==down&&realPath2(position(i,1)+1,position(i,2))~=0

position(i+1,:)=[position(i,1)+1,position(i,2)];

elseif max([right,up,left])==right&&realPath2(position(i,1),position(i,2)+1)~=0

position(i+1,:)=[position(i,1),position(i,2)+1];

elseif max([up,left])==up&&realPath2(position(i,1)-1,position(i,2))~=0

position(i+1,:)=[position(i,1)-1,position(i,2)];

elseif realPath2(position(i,1),position(i,2)-1)~=0

position(i+1,:)=[position(i,1),position(i,2)-1];

else

count=count+1;

position(i+1,:)=position(i-count,:);

count=count+1;

end

realPath1(position(i,1),position(i,2))=0;

% elseif i

elseif i>707

position(i+1,:)=[position(i,1)-1,position(i,2)];

elseif i>687

position(i+1,:)=[position(i,1),position(i,2)-1];

elseif i>668

position(i+1,:)=[position(i,1)+1,position(i,2)];

elseif i>587

position(i+1,:)=[position(i,1),position(i,2)-1];

else

position(i+1,:)=[position(i,1),position(i,2)+1];

end

plot(position(i,2)\*50,(102-position(i,1))\*50+5000,'r.','MarkerSize',9)

concentration(i)=C(min(position(i,1)-1,100),min(position(i,2)-1,100));

end

set(gca,'FontSize',14)

xlabel('X (m)')

ylabel('Y (m)')

figure

plot([1:i]/10\*3,concentration)

set(gca,'FontSize',14)

xlabel('time (min)')

figure

plot([1:i]/10\*3,cumsum(concentration))

set(gca,'FontSize',14)

xlabel('time (min)')

ylabel('C (ppm)')

figure;hold on

plot([100 100],[0 5000],'k','LineWidth',1)

plot([100 5100],[5000 5000],'k','LineWidth',1)

plot([100 5100],[50 50],'k','LineWidth',1)

plot([1100 1100],[0 5000],'k','LineWidth',1)

plot([2100 2100],[0 5000],'k','LineWidth',1)

plot([3100 3100],[0 5000],'k','LineWidth',1)

plot([4100 4100],[0 5000],'k','LineWidth',1)

plot([2100 5100],[4000 4000],'k','LineWidth',1)

plot([1100 5100],[1000 1000],'k','LineWidth',1)

plot([1100 2100],[2500 2500],'k','LineWidth',1)

xlim([0,5200])

ylim([0,5200])

% position=zeros(vMax\*time/50+1,2);

position1(1,:)=[2 2];

for i=1:vMax\*time/50

down=rand;

right=rand;

up=rand;

left=rand;

if max([down,right,up,left])==down&&realPath2(position1(i,1)+1,position1(i,2))~=0

position1(i+1,:)=[position1(i,1)+1,position1(i,2)];

elseif max([right,up,left])==right&&realPath2(position1(i,1),position1(i,2)+1)~=0

position1(i+1,:)=[position1(i,1),position(i,2)+1];

elseif max([up,left])==up&&realPath2(position1(i,1)-1,position1(i,2))~=0

position1(i+1,:)=[position1(i,1)-1,position1(i,2)];

elseif realPath2(position1(i,1),position1(i,2)-1)~=0

position1(i+1,:)=[position1(i,1),position1(i,2)-1];

elseif realPath2(position1(i,1)+1,position1(i,2))~=0

position1(i+1,:)=[position1(i,1)+1,position1(i,2)];

elseif realPath2(position1(i,1),position1(i,2)+1)~=0

position1(i+1,:)=[position1(i,1),position(i,2)+1];

end

plot(position1(i,2)\*50,102-position1(i,1)\*50+5000,'r.','MarkerSize',9)

concentration1(i)=C(min(position1(i,1)-1,100),min(position1(i,2)-1,100));

end

set(gca,'FontSize',14)

xlabel('X (m)')

ylabel('Y (m)')

figure

plot([1:i]/10\*3,concentration1)

set(gca,'FontSize',14)

xlabel('time (min)')

figure

plot([1:i]/10\*3,cumsum(concentration1))

set(gca,'FontSize',14)

xlabel('time (min)')

ylabel('C (ppm)')

clc

clear

close all

vMax=10000;

time=4;

load('position\_4.mat');

load('C\_10\_leaking.mat');

for i=1:vMax\*time/50

concentration(i)=C(min(position(i,1)-1,100),min(position(i,2)-1,100));

end

figure

plot([1:i]/10\*3,concentration)

set(gca,'FontSize',14)

xlabel('time (min)')

figure

plot([1:i]/10\*3,cumsum(concentration))

set(gca,'FontSize',14)

xlabel('time (min)')

ylabel('C (ppm)')

**Reference**

[1] S. Ferrari, *MAE 6790: Final Project*