CSE571_HW2

1.3 A* Implementation

Results:

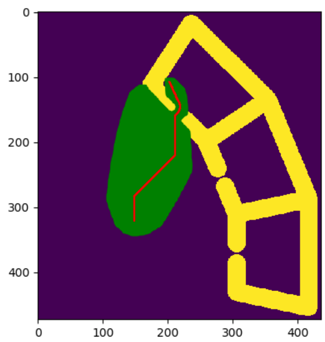


Fig 1. ε =1.

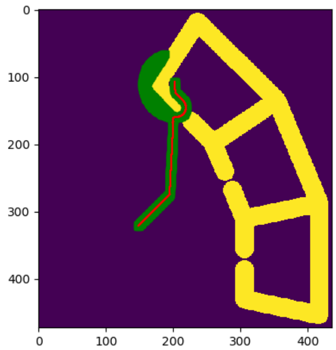


Fig 2. ε =10.

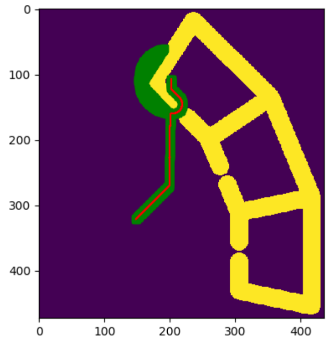


Fig 3. ε =20.

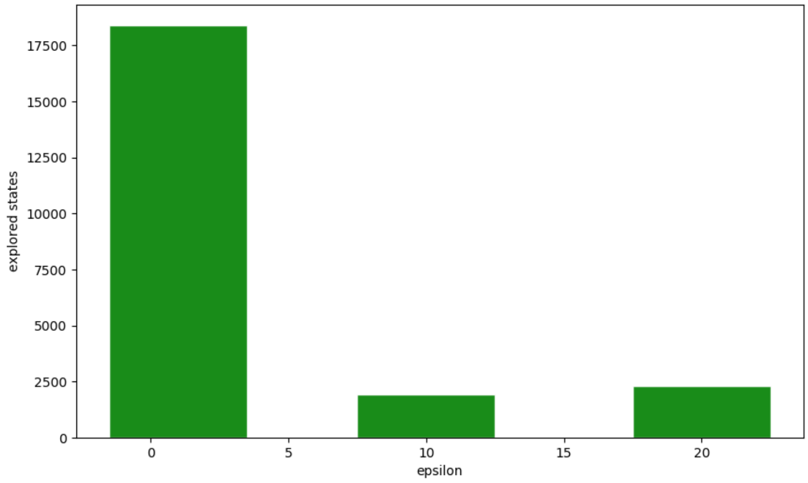


Fig 4. Number of states explored under different ε .

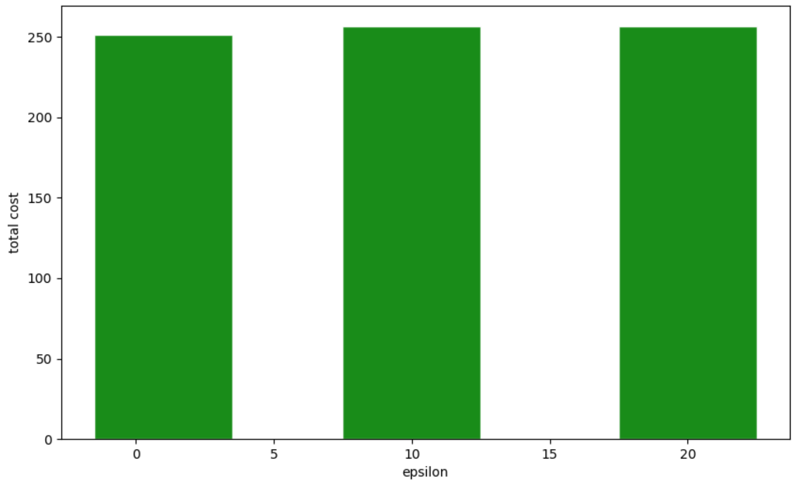


Fig 5. Total cost under different ε .

Table 1. A* search results under different value of ε .

A* search	ε = 1	ε = 10	ε = 20
Explored states	18406	1935	2330
Total cost	251.432	256.7	256.7

Discussion:

From fig.1 to fig.5, I found that in a* search, when I increase the ε , the number of explored states will decrease. But after ε is higher than critical value, the explored states will still increase. I think it is because when increasing the ε , in some place in the map that place is closer to the goal but there is still a wall between them. And because it more focuses on the distance between current position and goal instead of start point, thus he can keep searching new place which is closer to goal before he reaches the goal, and thus, increased the number of explored states. Even though, the number of explored states may increase, when compared to the original states number, the number of states reduction is still magnificent to cover the drawback.

I also find the that the total cost may increase when ε is higher than one. This means the path it searched is not the shortest path.

And I also tried to decrease the ε to be less than 1 or be 0, and realized that if I do this, it will become a kind of Dijkstra algorithm which guarantee the shortest path but take much more time to search. And when I compared the cost between Dijkstra and a* search with ε =1, I found that the costs are the same, which implies the ε =1 a* found the shortest path, but is much faster.

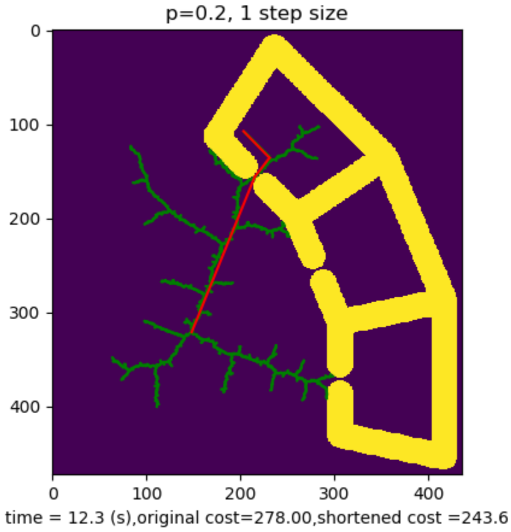
1.4

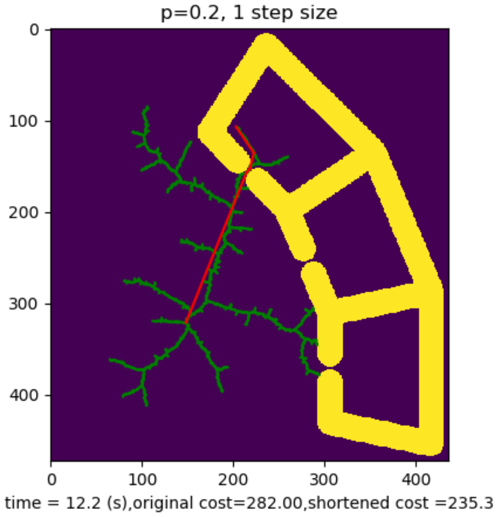
RRT:

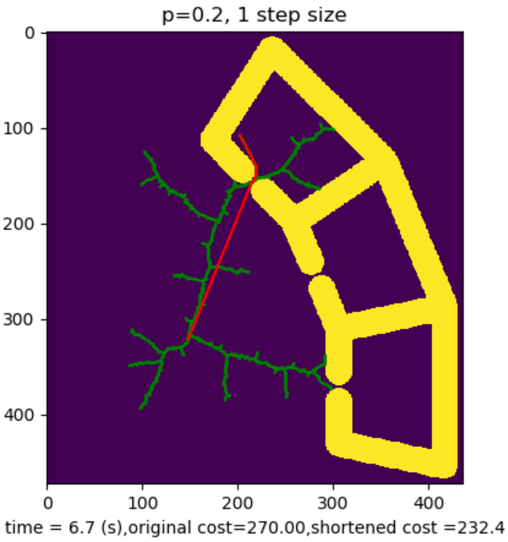
Results:

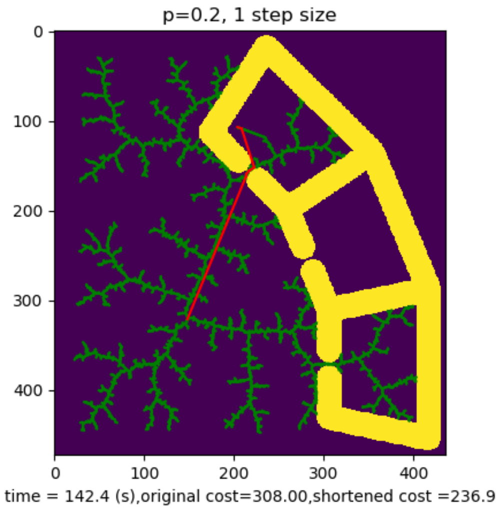
(a)

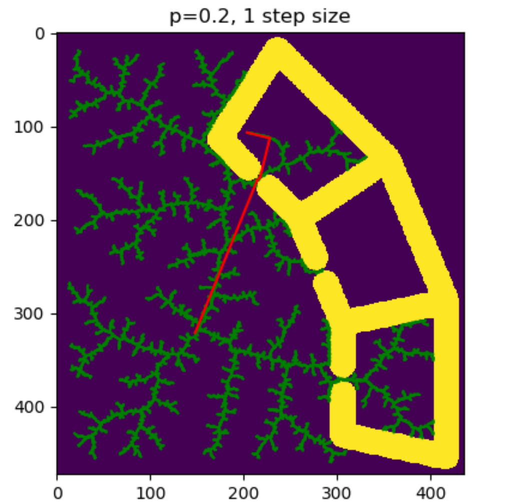
P=0.2, step size = 1



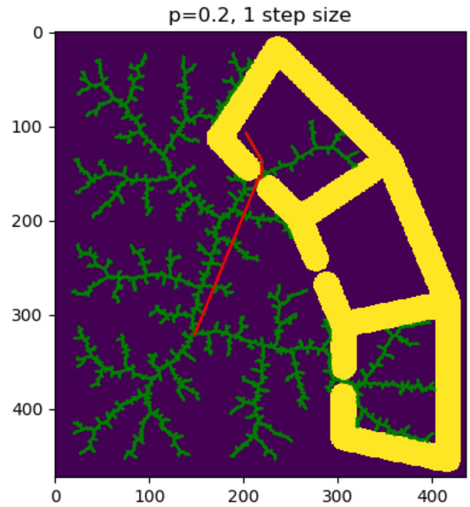




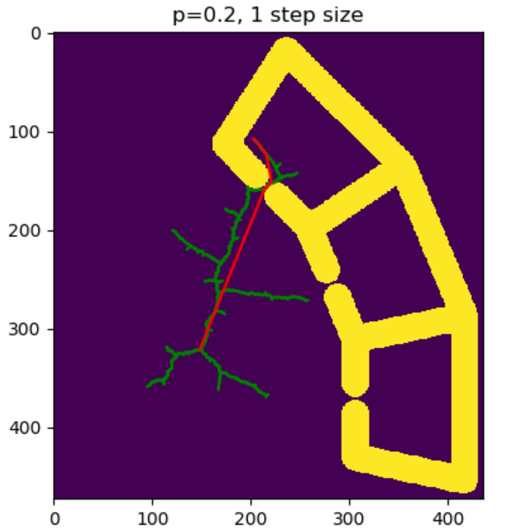




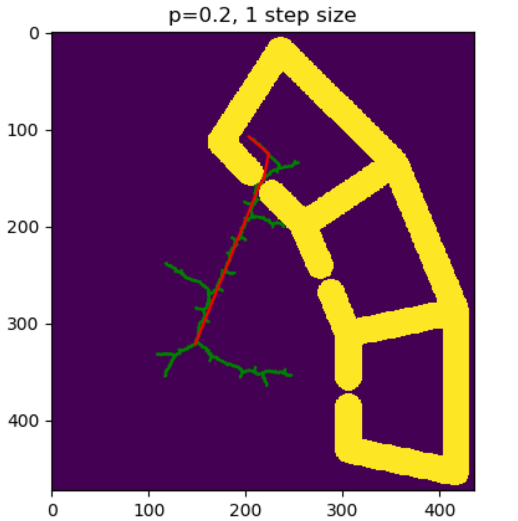
time = 259.7 (s),original cost=326.00,shortened cost =249.3



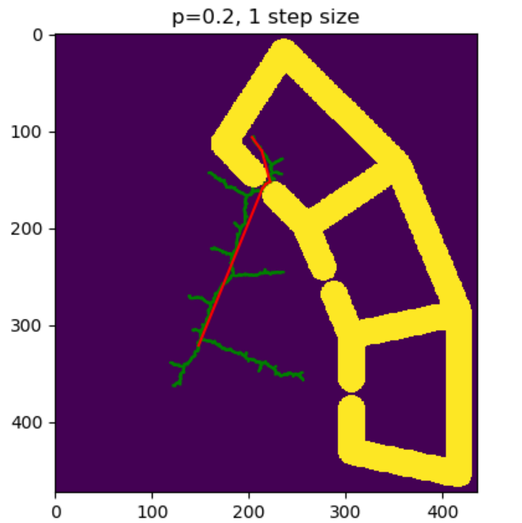
time = 193.1 (s),original cost=273.00,shortened cost =233.4



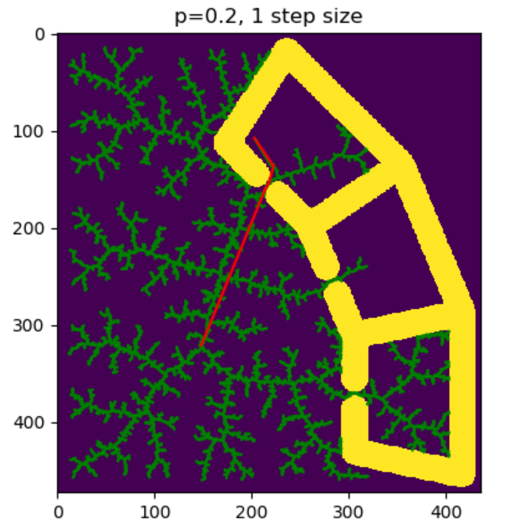
time = 1.2 (s),original cost=285.00,shortened cost =234.8



time = 2.1 (s),original cost=290.00,shortened cost =238.0



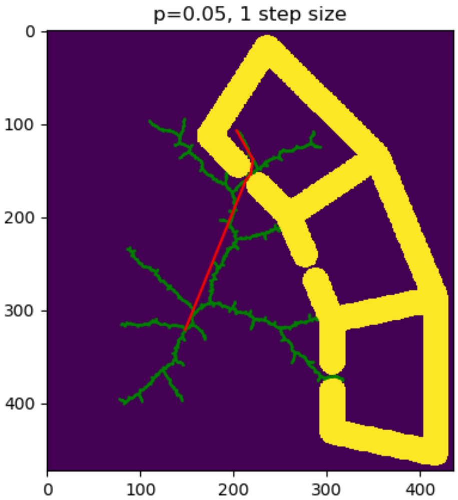
time = 1.1 (s),original cost=280.00,shortened cost =236.1



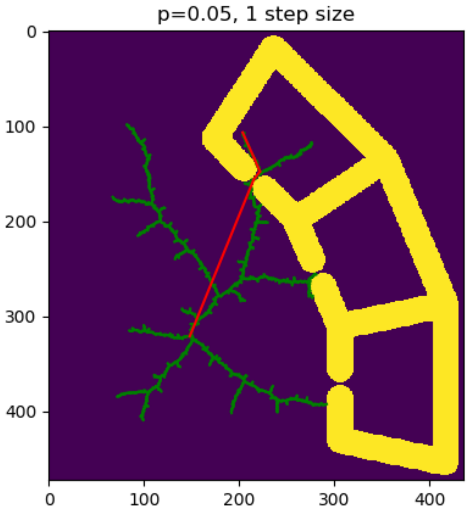
time = 380.4 (s),original cost=276.00,shortened cost =234.6

(b)

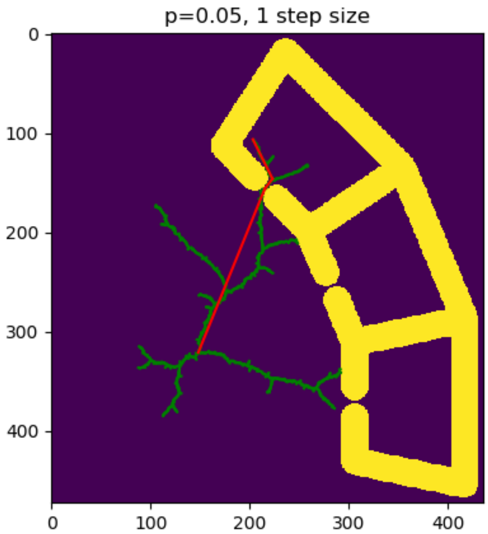
P=0.05, step size =1



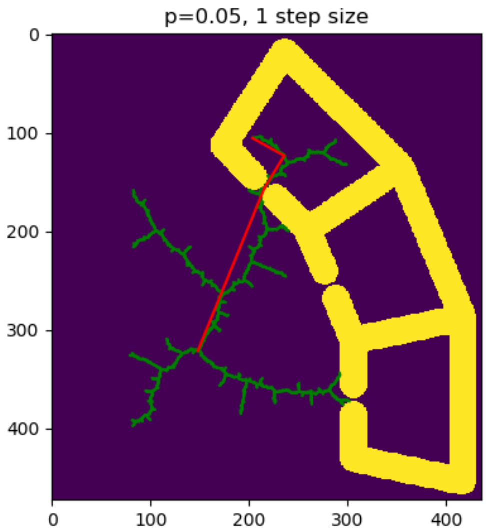
time = 3.0 (s),original cost=284.00,shortened cost =233.4



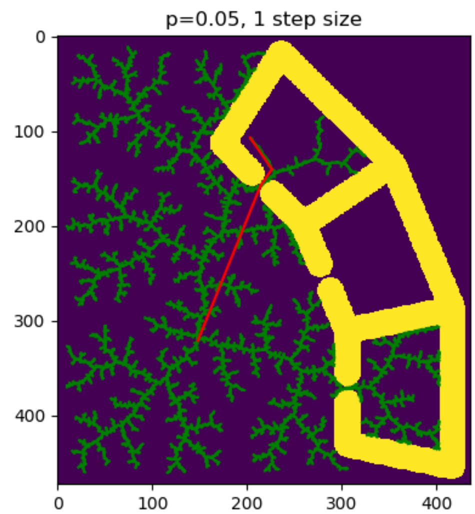
time = 4.1 (s),original cost=281.00,shortened cost =233.9



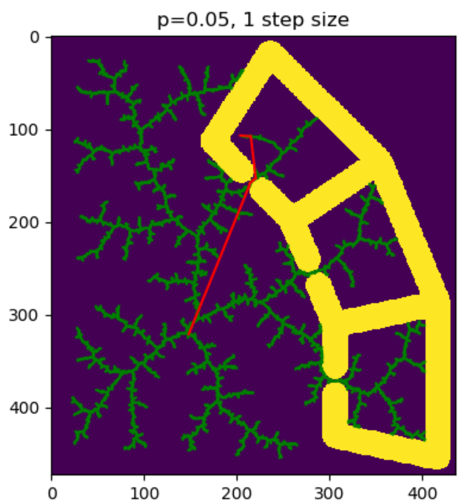
time = 1.5 (s),original cost=282.00,shortened cost =235.3



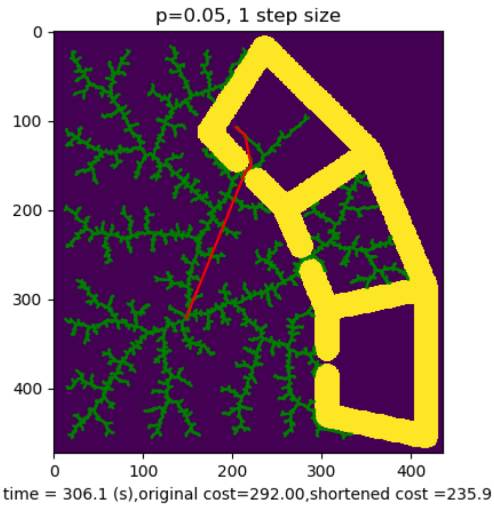
time = 5.2 (s),original cost=309.00,shortened cost =253.7

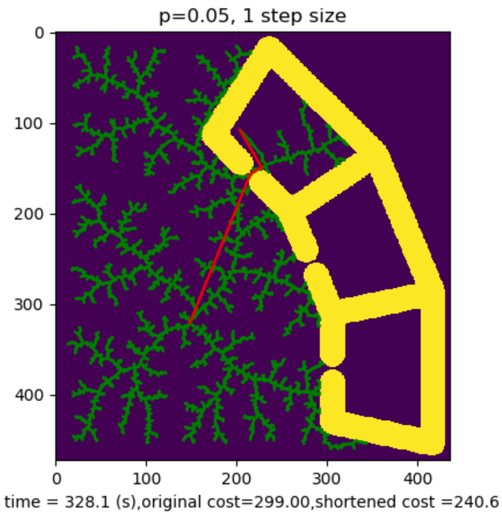


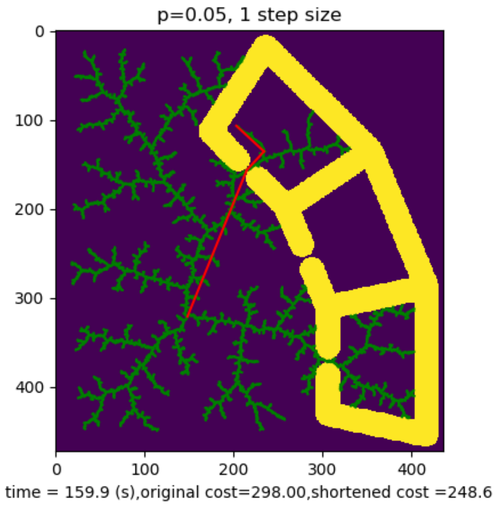
time = 230.7 (s),original cost=310.00,shortened cost =238.1

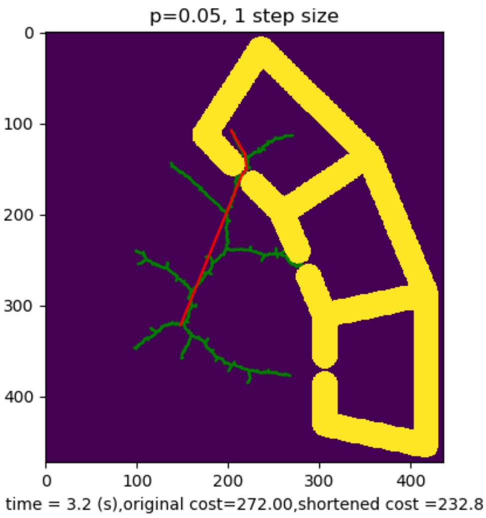


time = 162.5 (s),original cost=331.00,shortened cost =240.8


p=0.05, 1 step size
time = 306.1 (s),original cost=292.00,shortened cost =235.9

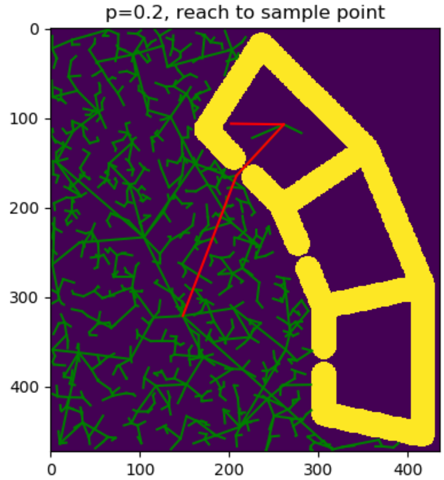

p=0.05, 1 step size
time = 328.1 (s),original cost=299.00,shortened cost =240.6


p=0.05, 1 step size
time = 159.9 (s),original cost=298.00,shortened cost =248.6

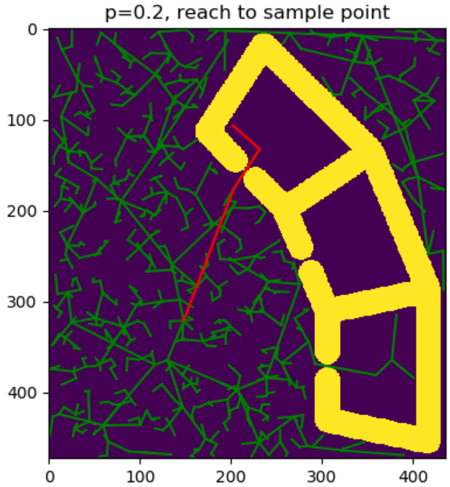

p=0.05, 1 step size
time = 3.2 (s),original cost=272.00,shortened cost =232.8

(c)

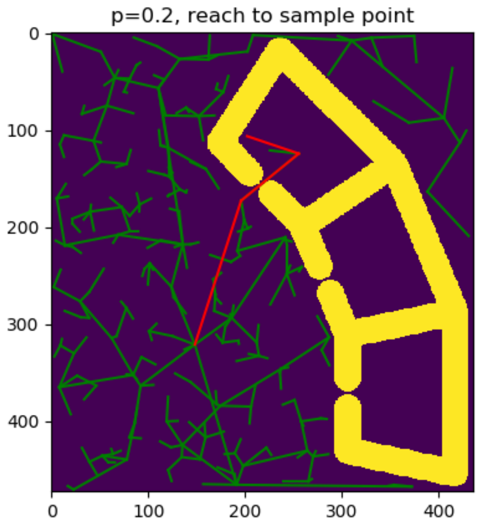
P=0.2, reach to sampled point



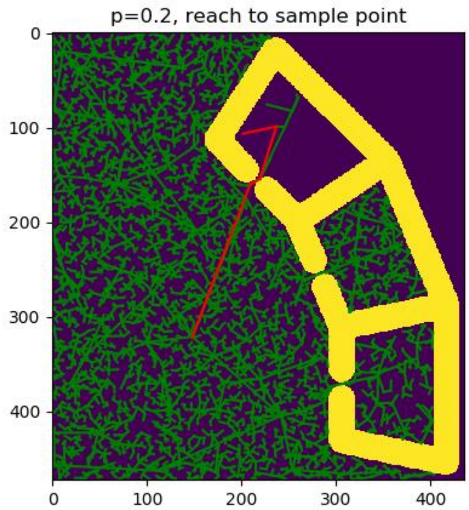
time = 1.2 (s),original cost=394.97,shortened cost =305.1



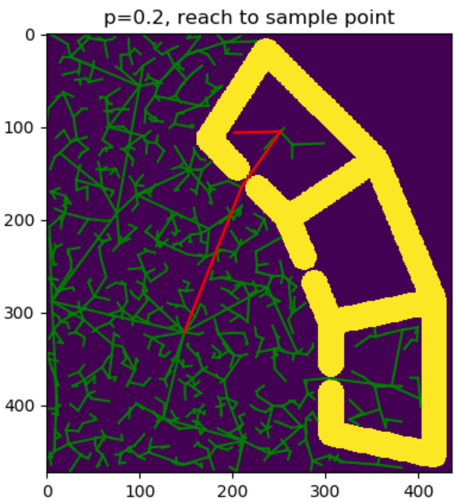
time = 0.7 (s),original cost=276.98,shortened cost =246.9



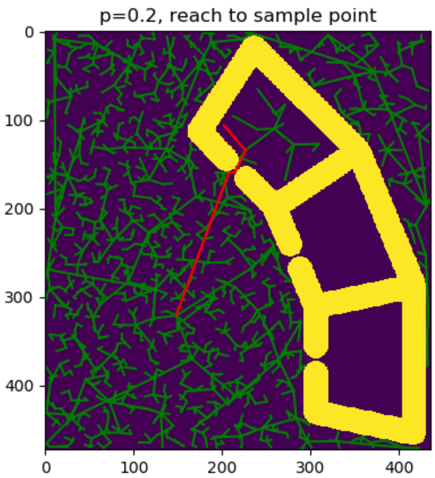
time = 1.0 (s),original cost=385.72,shortened cost =289.1



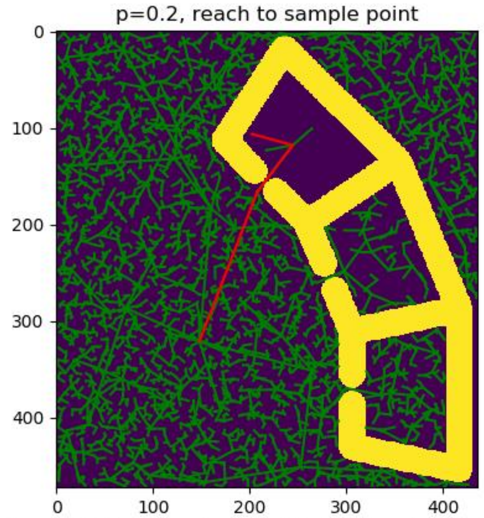
time = 38.4 (s),original cost=322.42,shortened cost =277.7



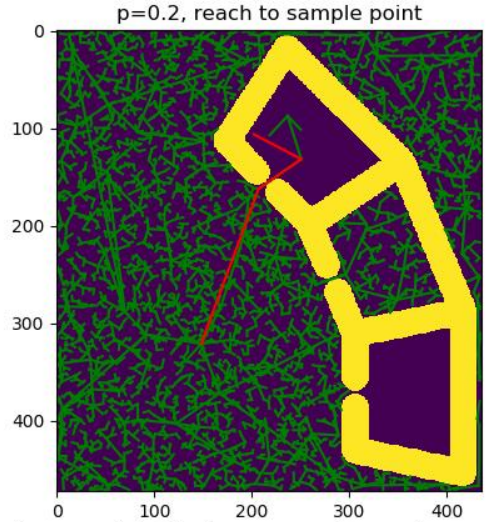
time = 1.6 (s),original cost=565.74,shortened cost =291.0



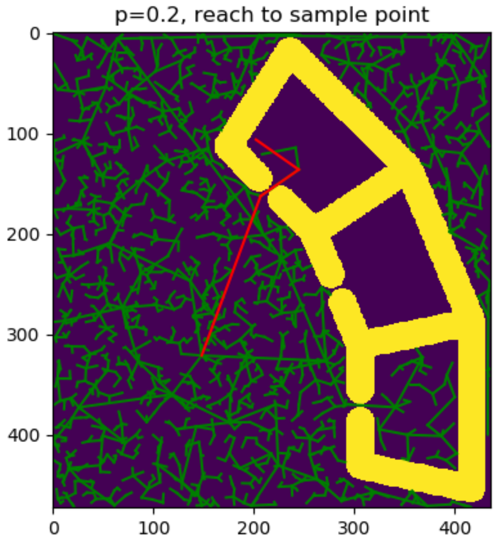
time = 5.6 (s),original cost=487.58,shortened cost =244.1



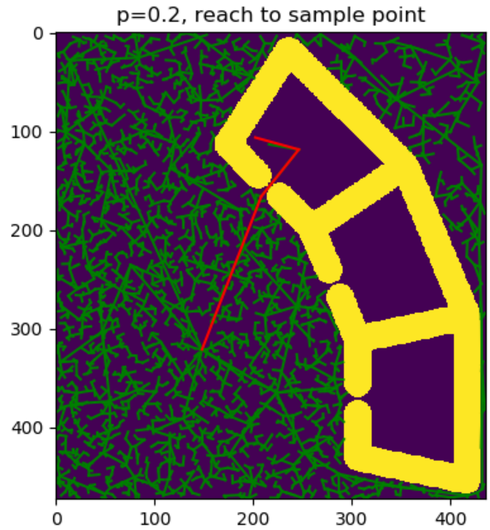
time = 26.6 (s),original cost=371.63,shortened cost =270.8



time = 29.0 (s),original cost=318.16,shortened cost =277.9



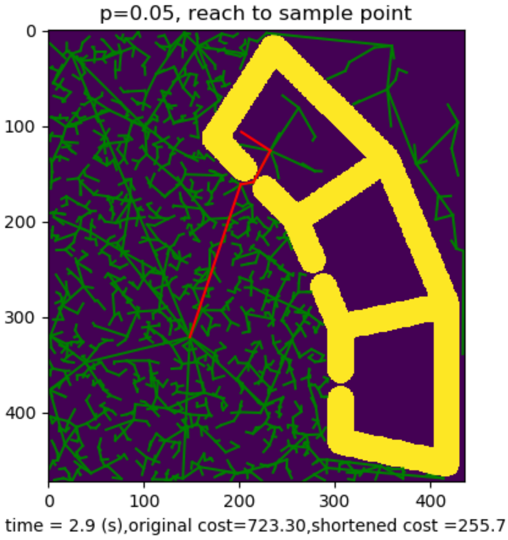
time = 7.7 (s),original cost=472.18,shortened cost =267.5

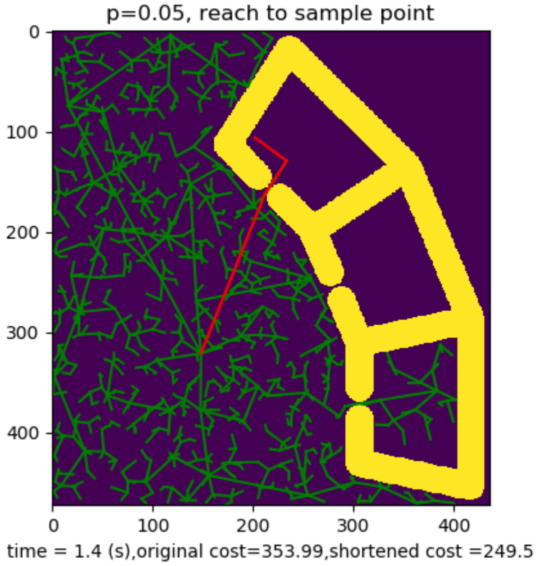


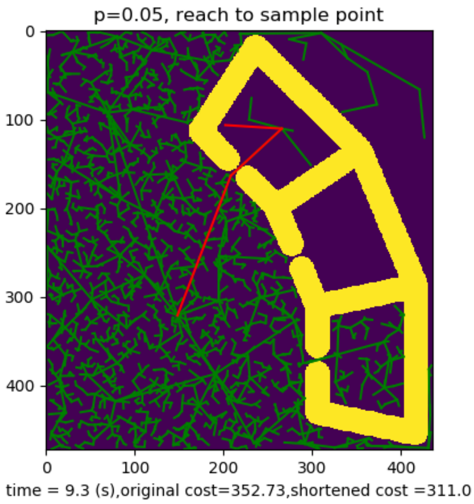
time = 20.2 (s),original cost=459.84,shortened cost =273.1

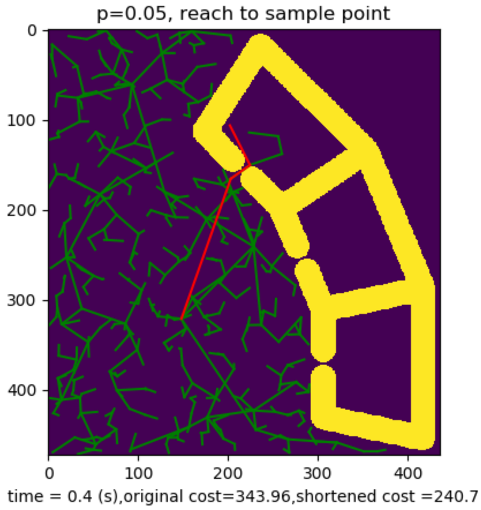
(d)

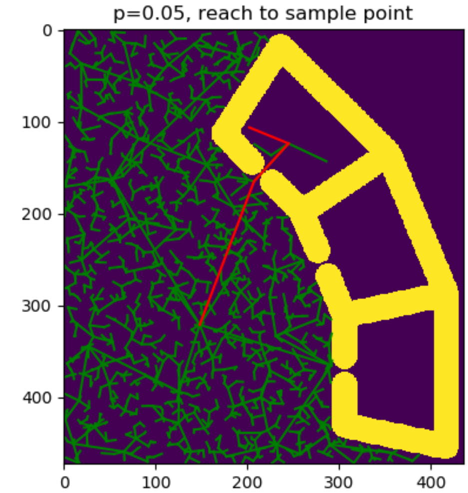
P=0.05, reach to sampled point



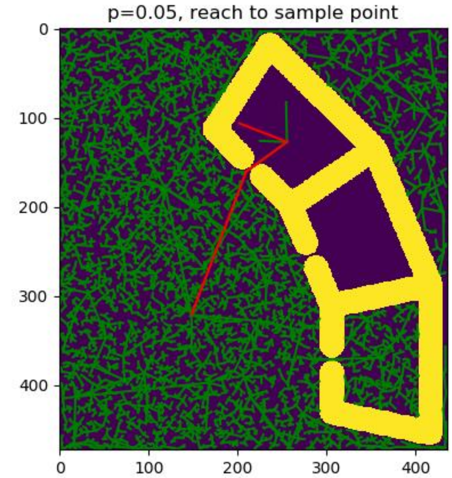




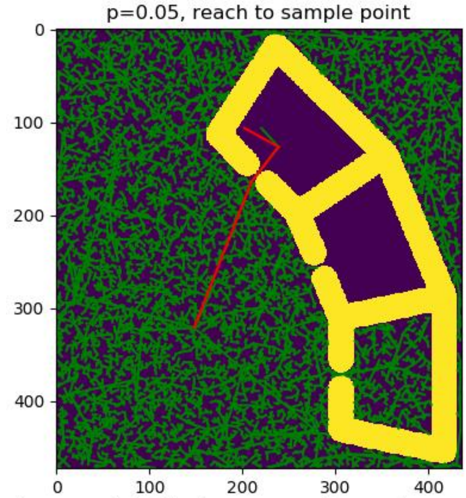




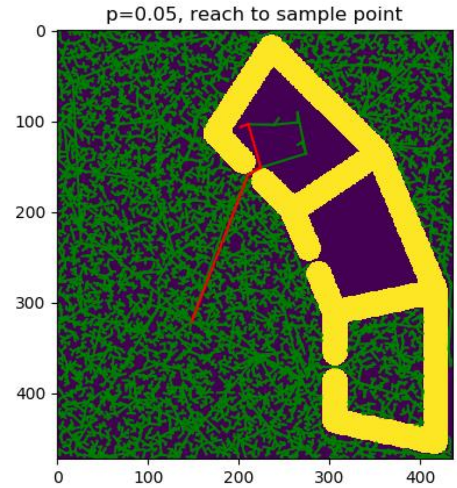
time = 3.3 (s),original cost=578.42,shortened cost =269.1



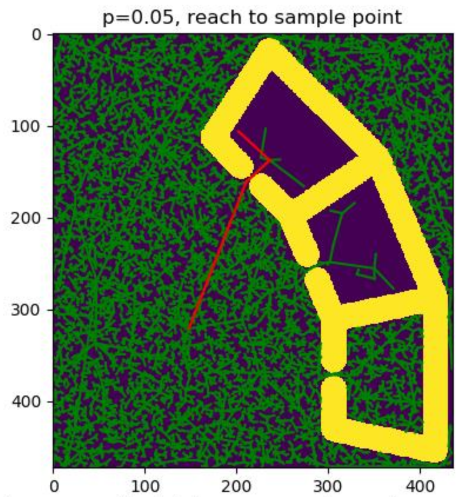
time = 58.6 (s),original cost=476.95,shortened cost =286.0



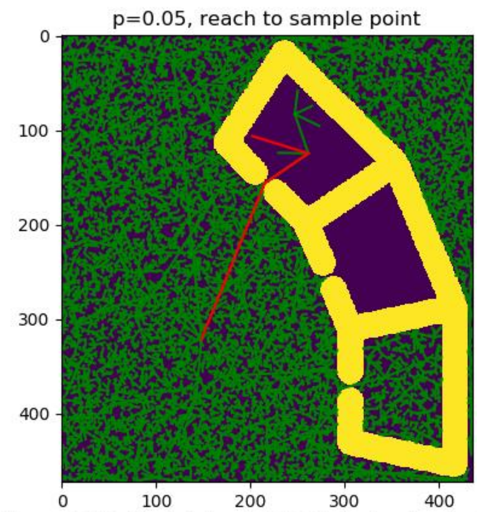
time = 87.2 (s),original cost=556.69,shortened cost =258.2



time = 116.0 (s),original cost=369.66,shortened cost =248.4



time = 121.8 (s),original cost=657.14,shortened cost =253.0

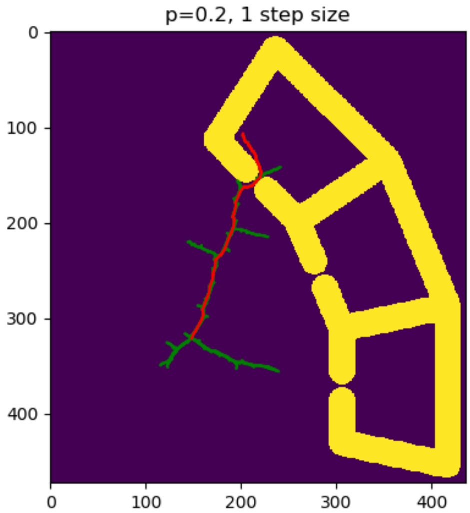


time = 197.9 (s),original cost=388.04,shortened cost =296.3

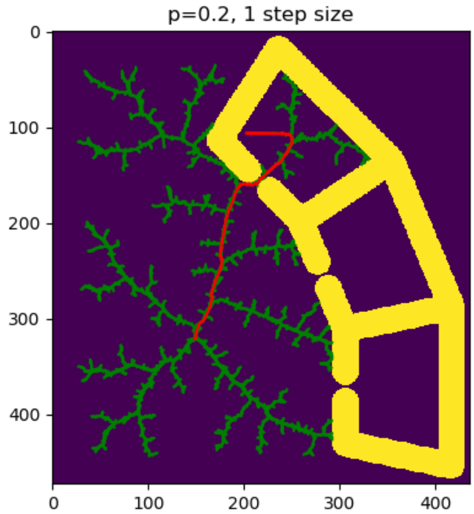
RRT*

(a)

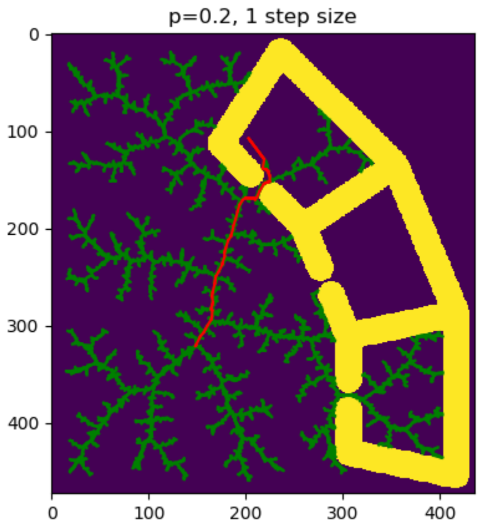
P = 0.2, 1 step size



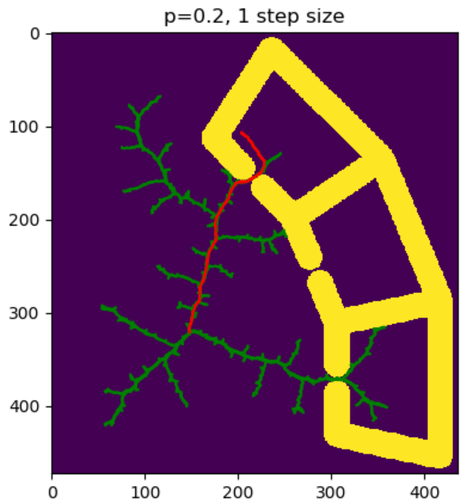
time = 1.2 (s),original cost=246.10



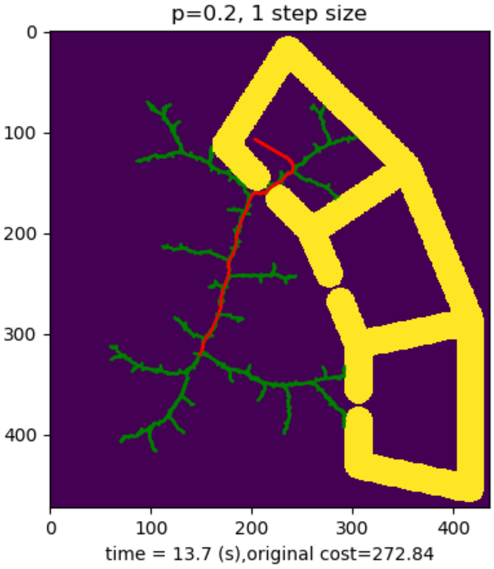
time = 122.6 (s),original cost=305.90

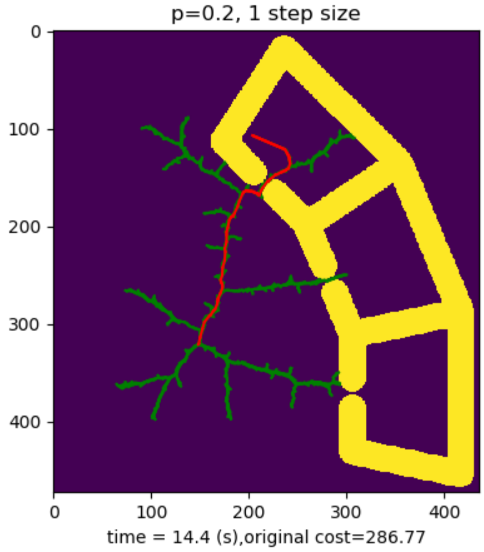


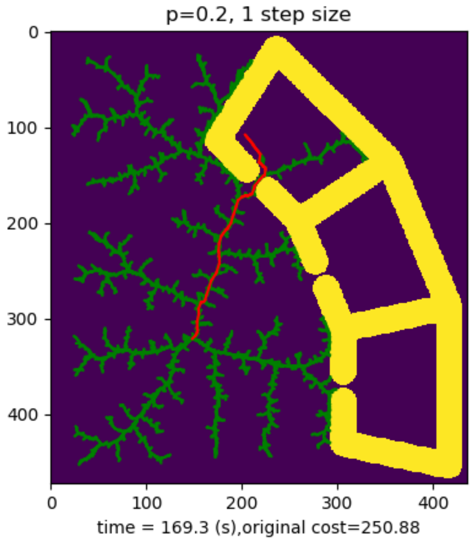
time = 293.4 (s),original cost=256.29

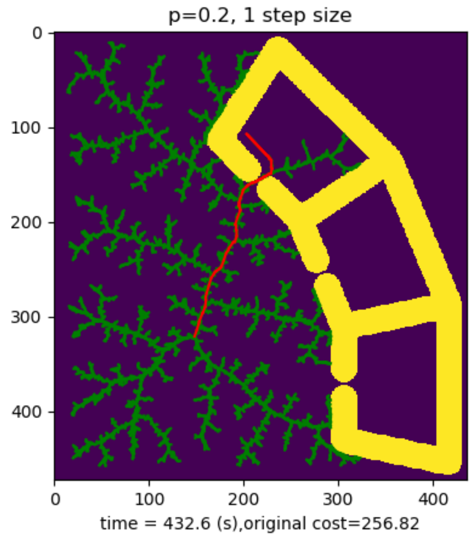


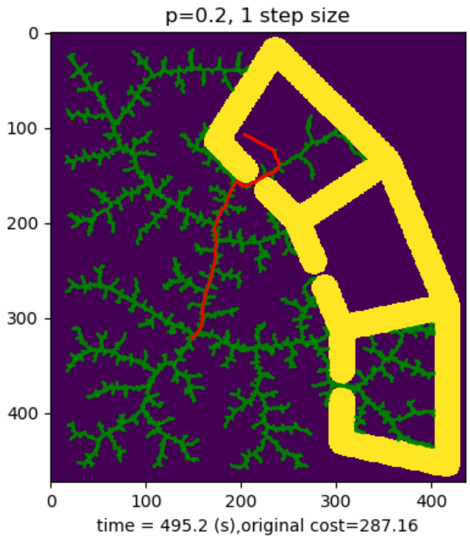
time = 15.0 (s),original cost=257.61

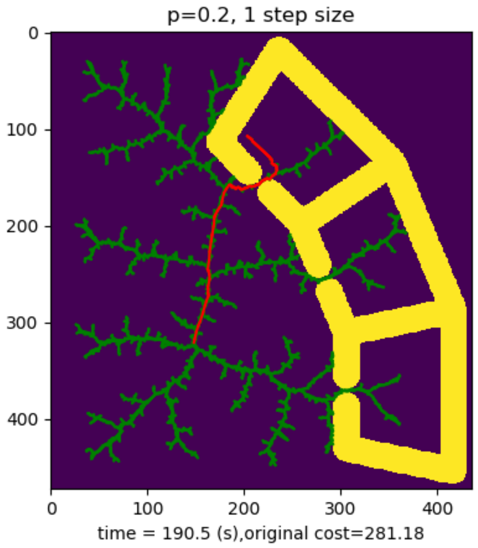

p=0.2, 1 step size
time = 13.7 (s),original cost=272.84


p=0.2, 1 step size
time = 14.4 (s),original cost=286.77


p=0.2, 1 step size
time = 169.3 (s),original cost=250.88

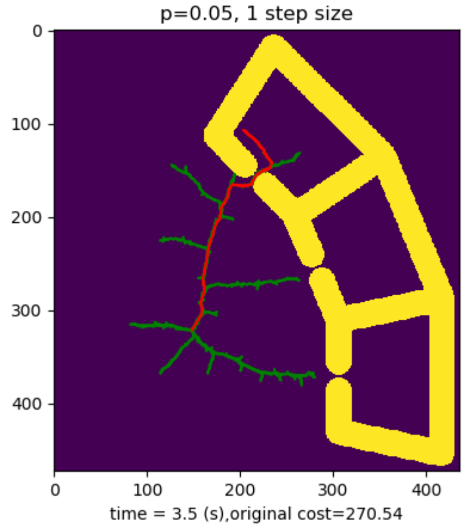

p=0.2, 1 step size
time = 432.6 (s),original cost=256.82

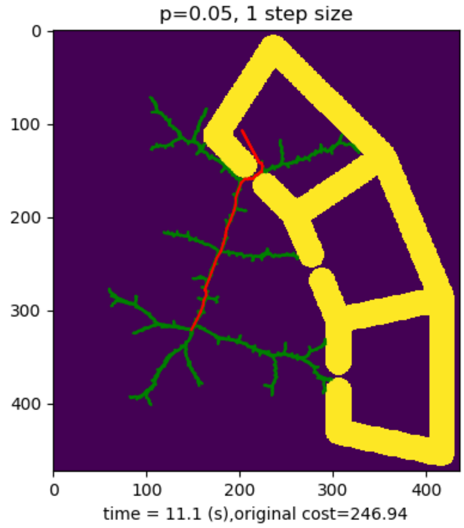

p=0.2, 1 step size
time = 495.2 (s),original cost=287.16

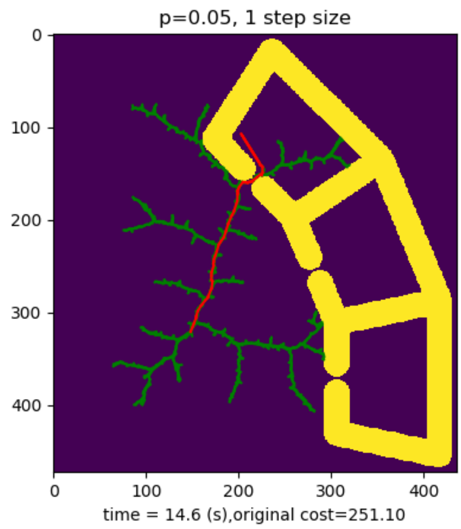

p=0.2, 1 step size
time = 190.5 (s),original cost=281.18

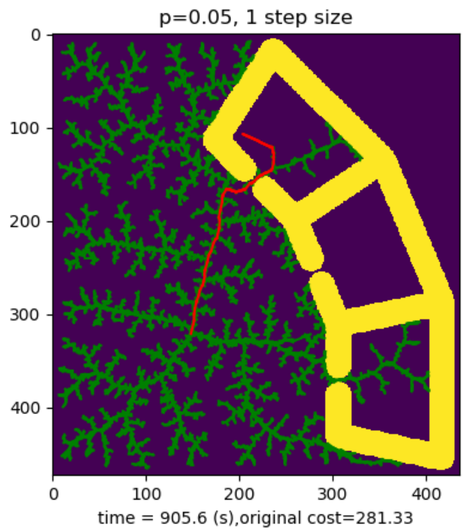
(b)

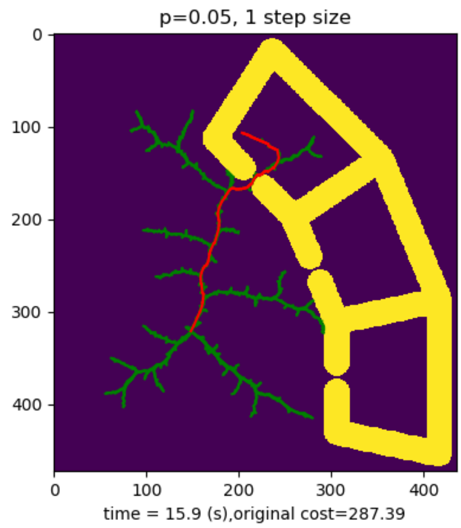
P=0.05, step size=1

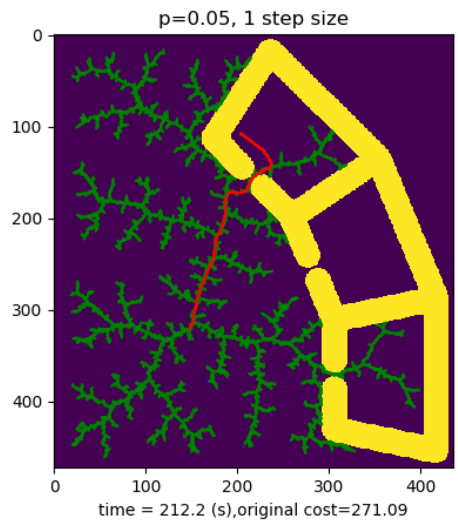


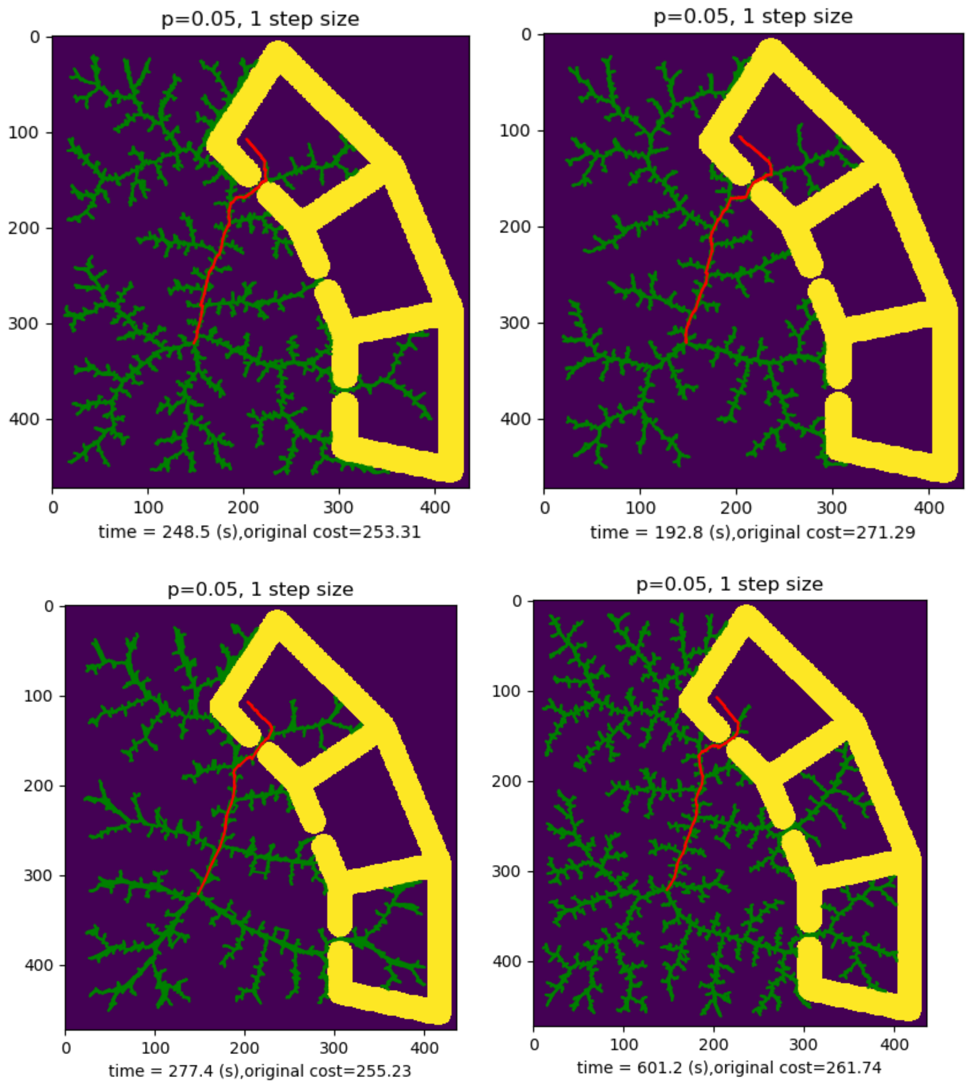






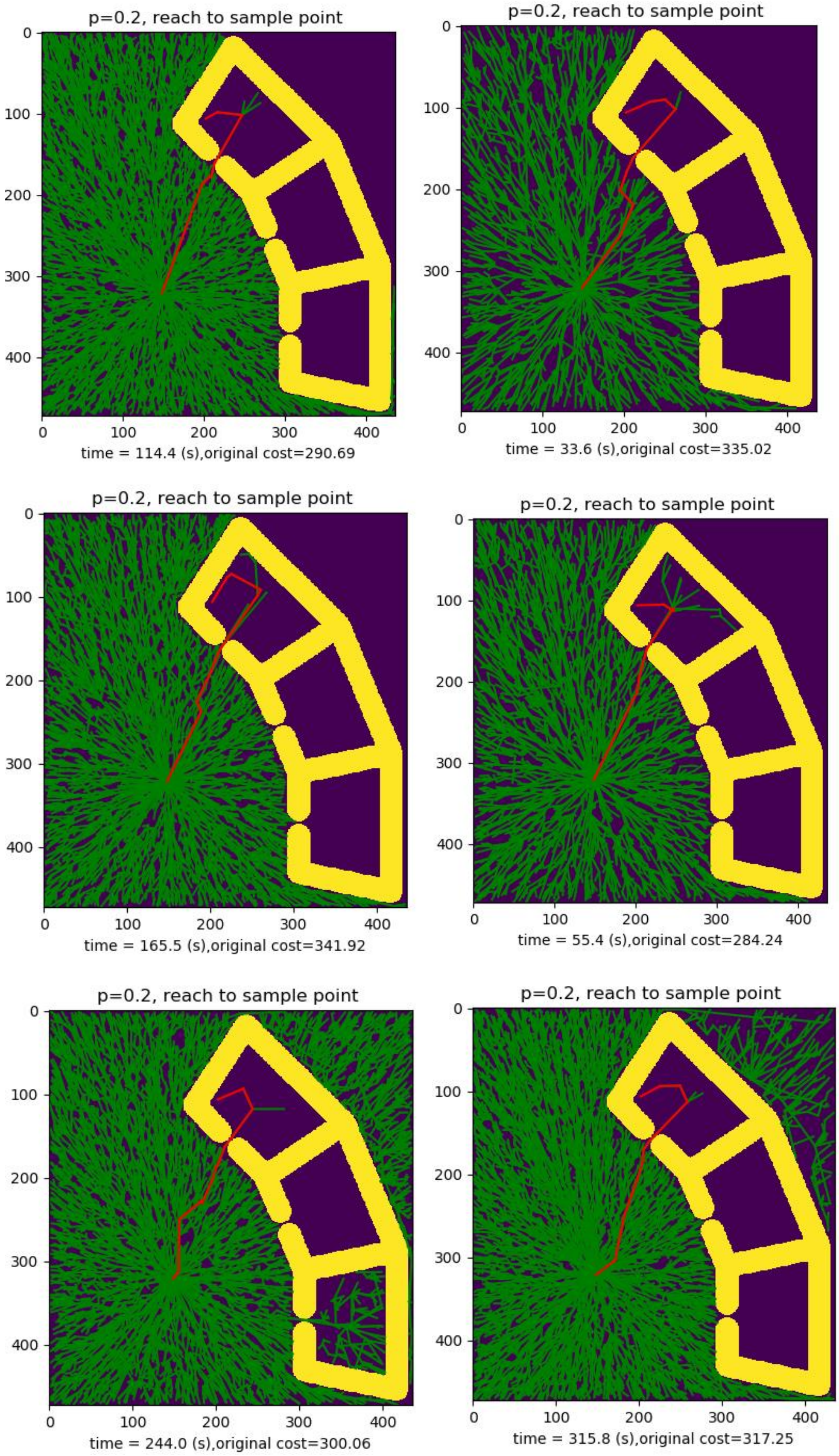


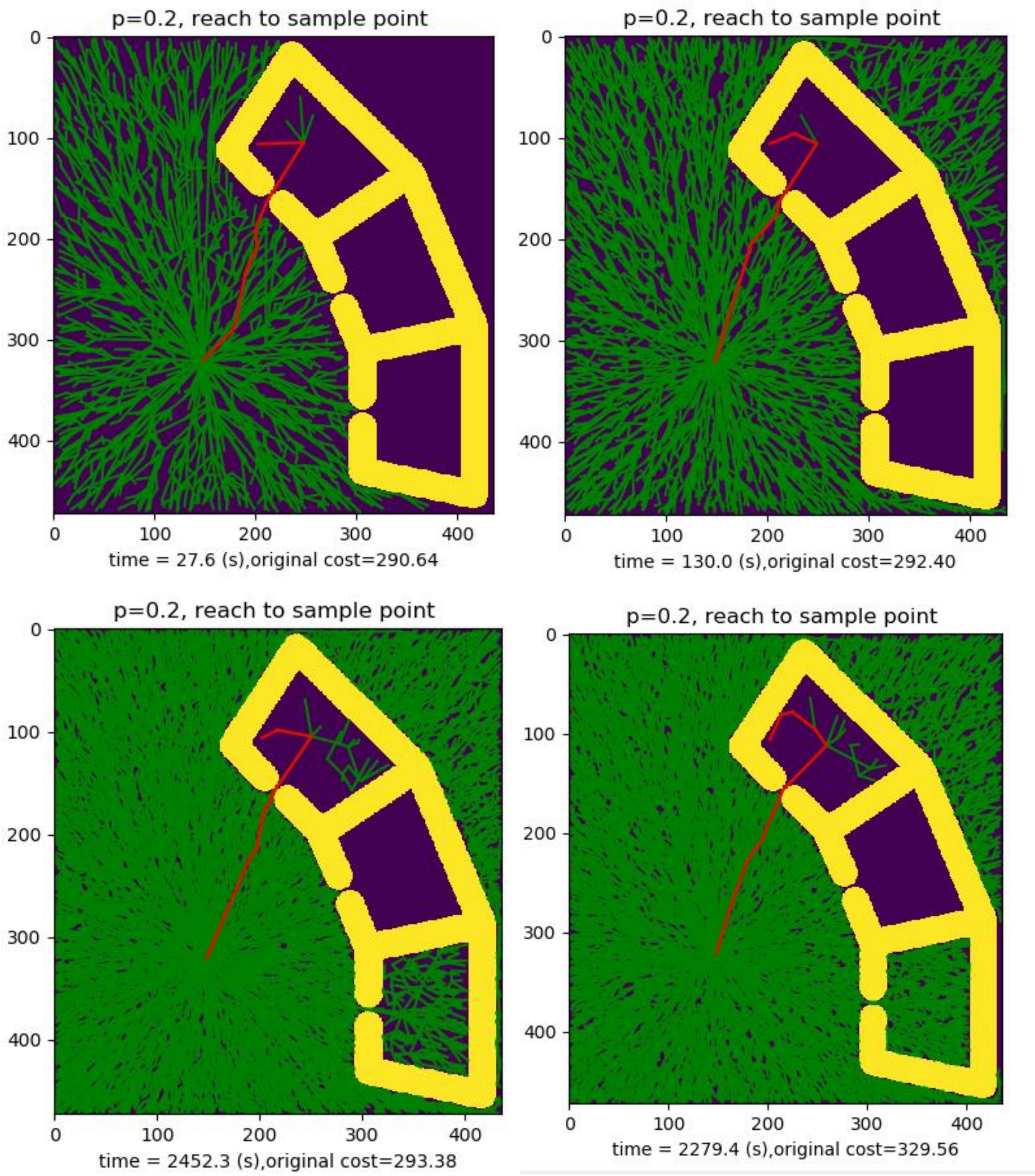



p=0.05, 1 step size
0
100
200
300
400
0
100
200
300
400
time = 248.5 (s),original cost=253.31
p=0.05, 1 step size
time = 192.8 (s),original cost=271.29
p=0.05, 1 step size
time = 277.4 (s),original cost=255.23
p=0.05, 1 step size
time = 601.2 (s),original cost=261.74

(c)

P=0.2, Reach to sample point

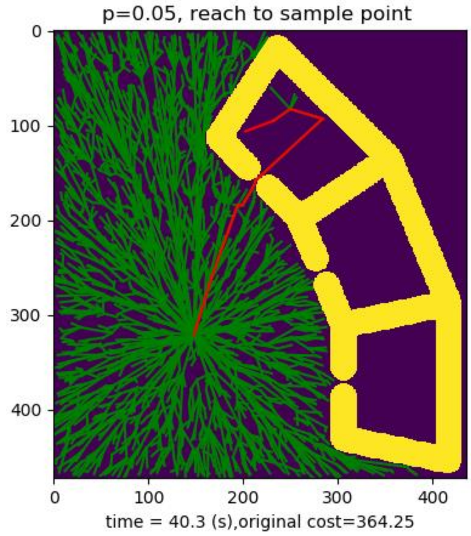


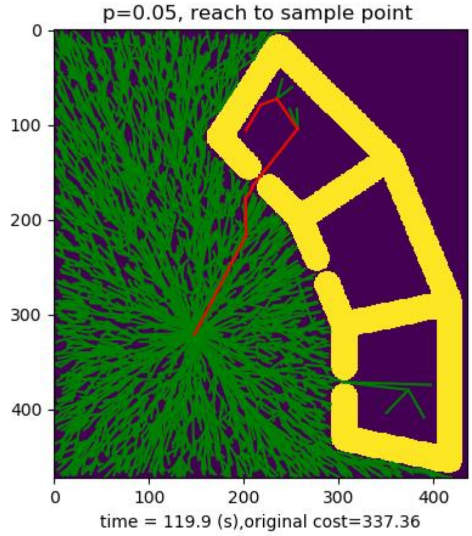


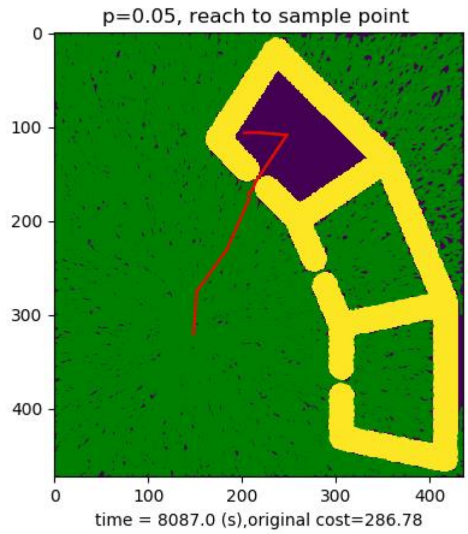
p=0.2, reach to sample point
time = 27.6 (s),original cost=290.64
p=0.2, reach to sample point
time = 130.0 (s),original cost=292.40
p=0.2, reach to sample point
time = 2452.3 (s),original cost=293.38
p=0.2, reach to sample point
time = 2279.4 (s),original cost=329.56

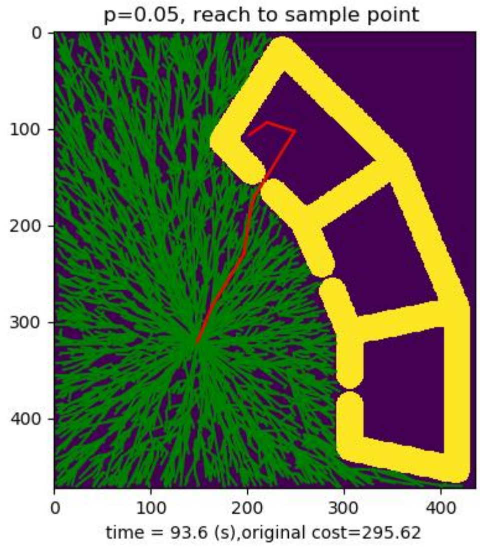
(d)

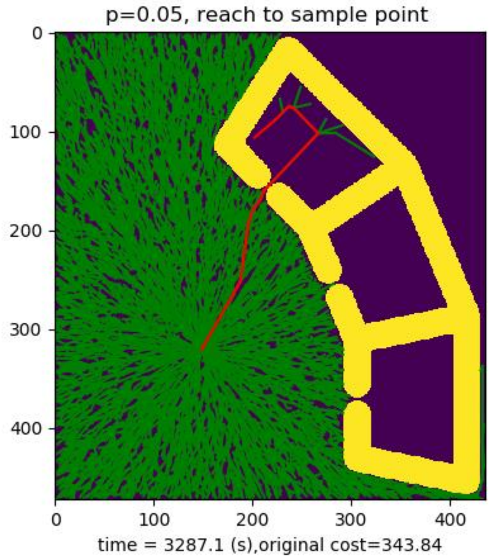
P = 0.05, reach to sample point

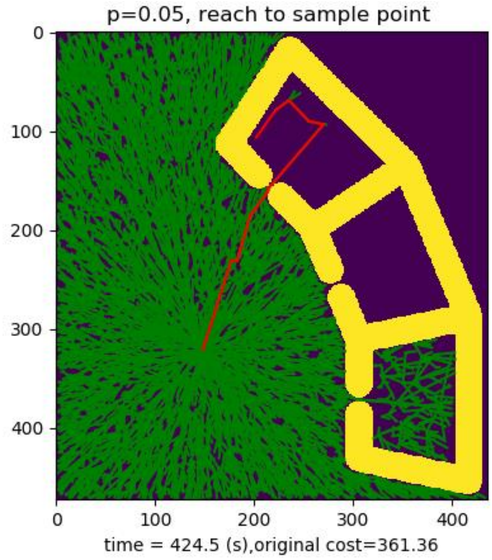


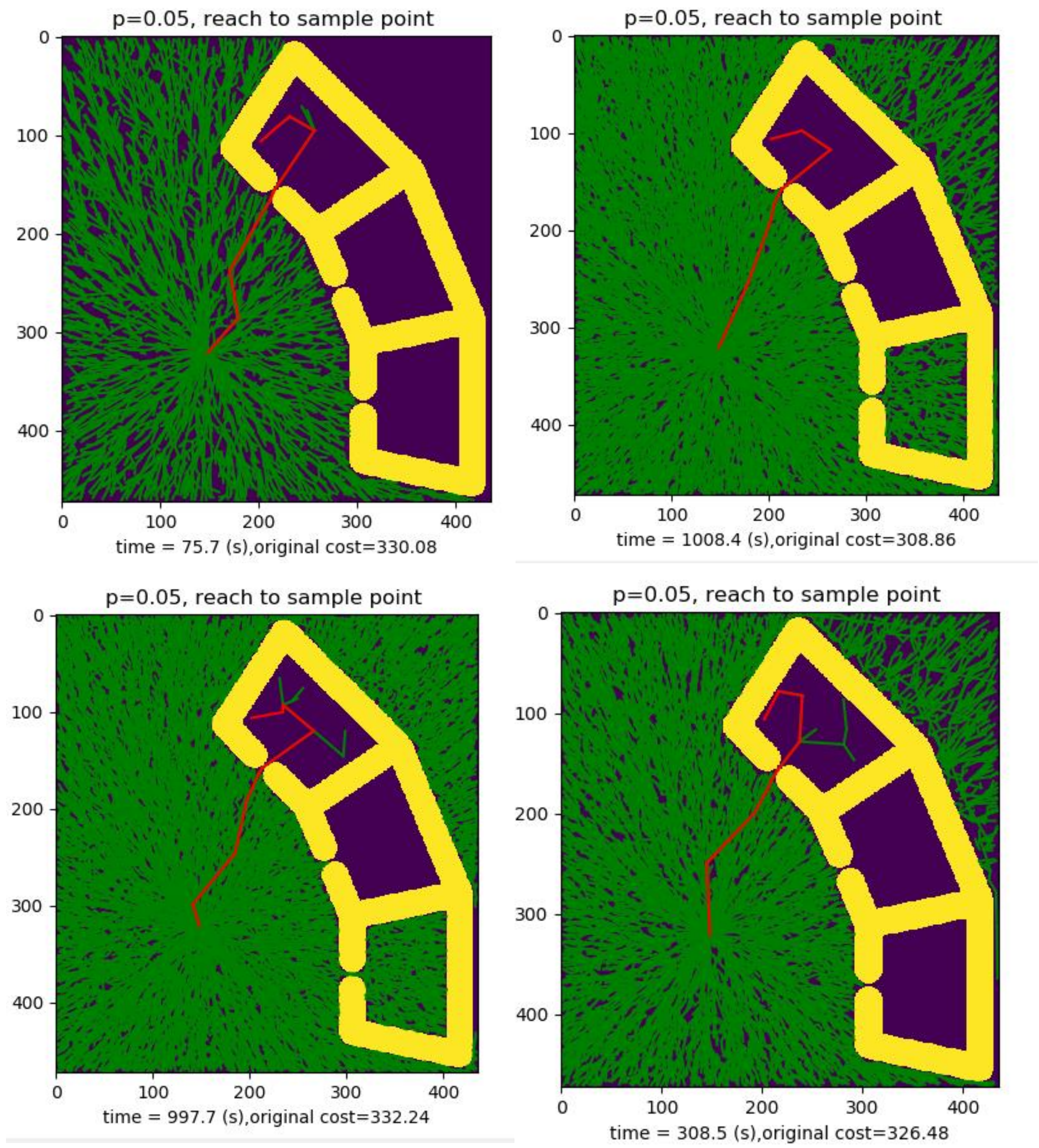











p=0.05, reach to sample point
0
100
200
300
400
0
100
200
300
400
time = 75.7 (s),original cost=330.08
p=0.05, reach to sample point
0
100
200
300
400
0
100
200
300
400
time = 1008.4 (s),original cost=308.86
p=0.05, reach to sample point
0
100
200
300
400
0
100
200
300
400
time = 997.7 (s),original cost=332.24
p=0.05, reach to sample point
0
100
200
300
400
0
100
200
300
400
time = 308.5 (s),original cost=326.48

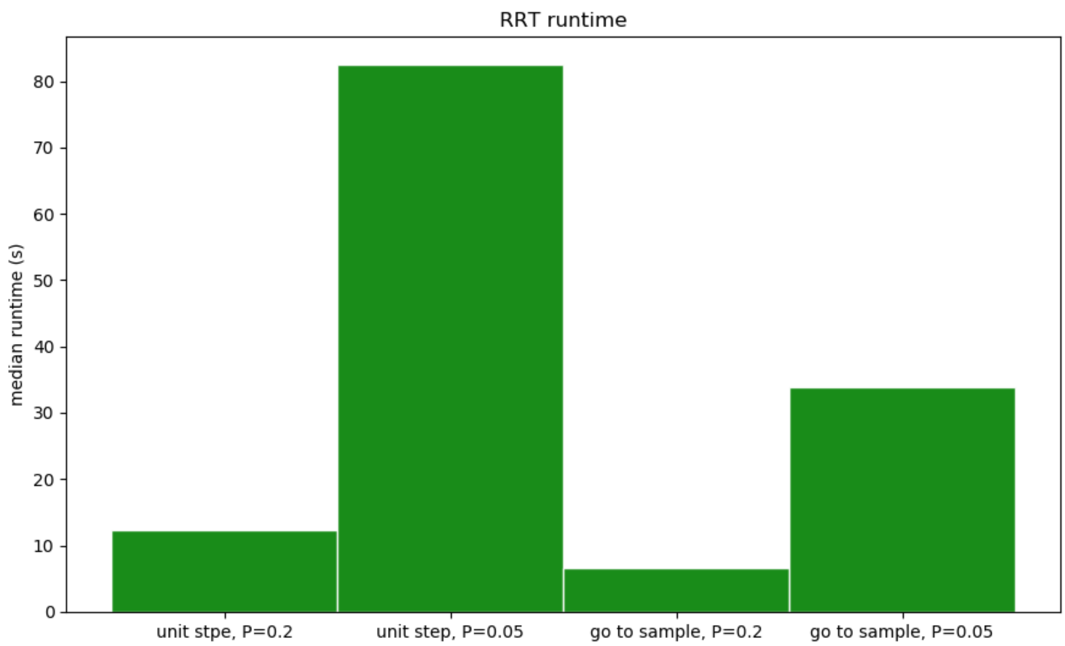


Fig 6. Runtime for different settings of RRT.

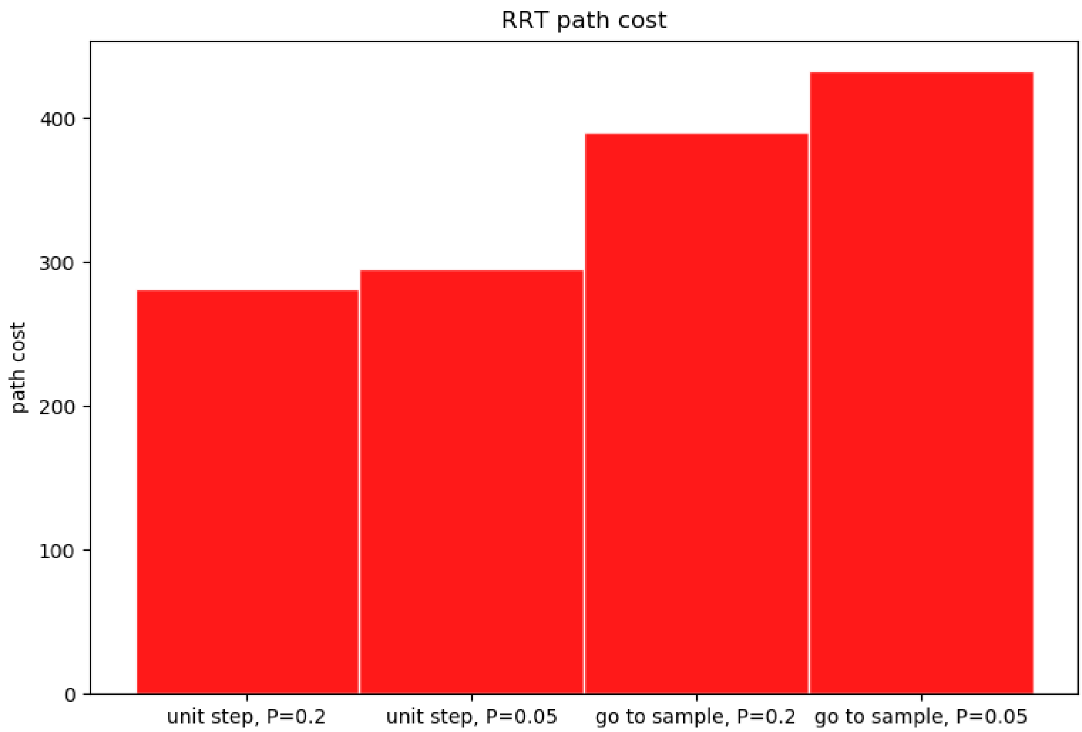


Fig 7. Path cost for different settings of RRT.

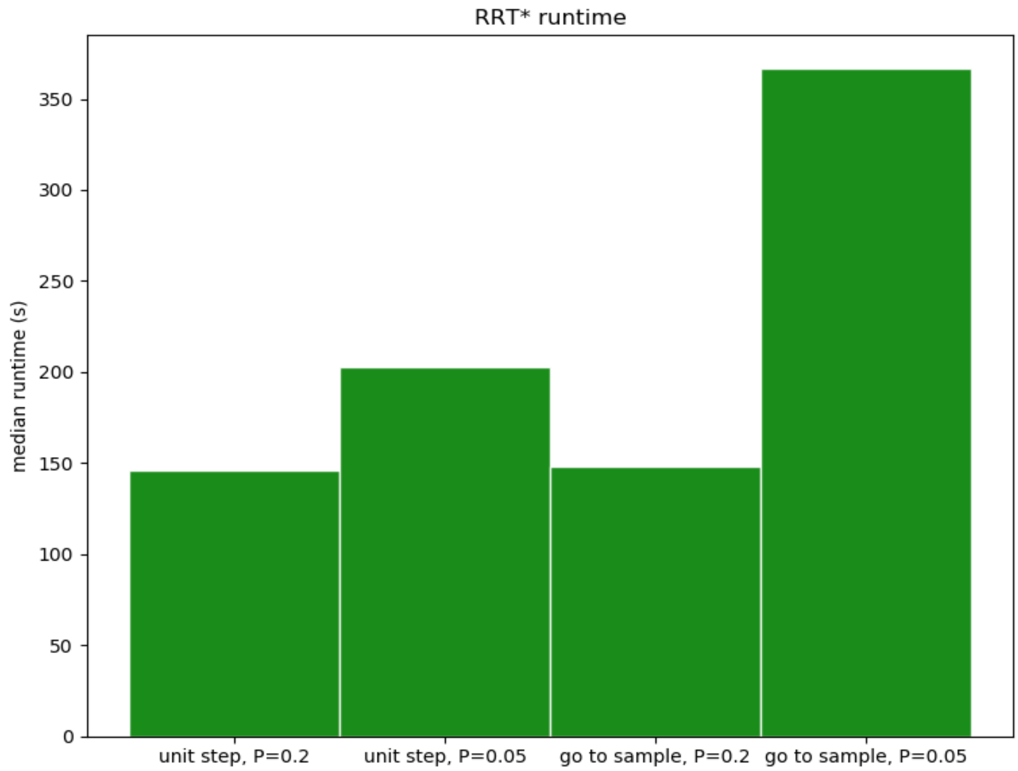


Fig 8. Runtime for different settings of RRT*.

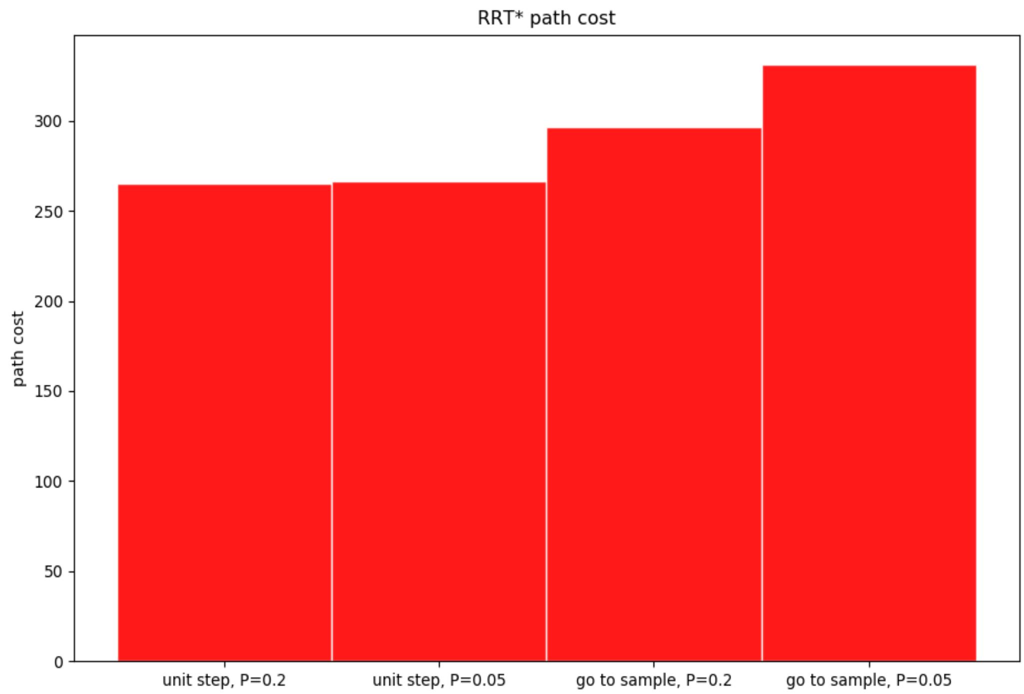


Fig 9. Path cost for different settings of RRT*.

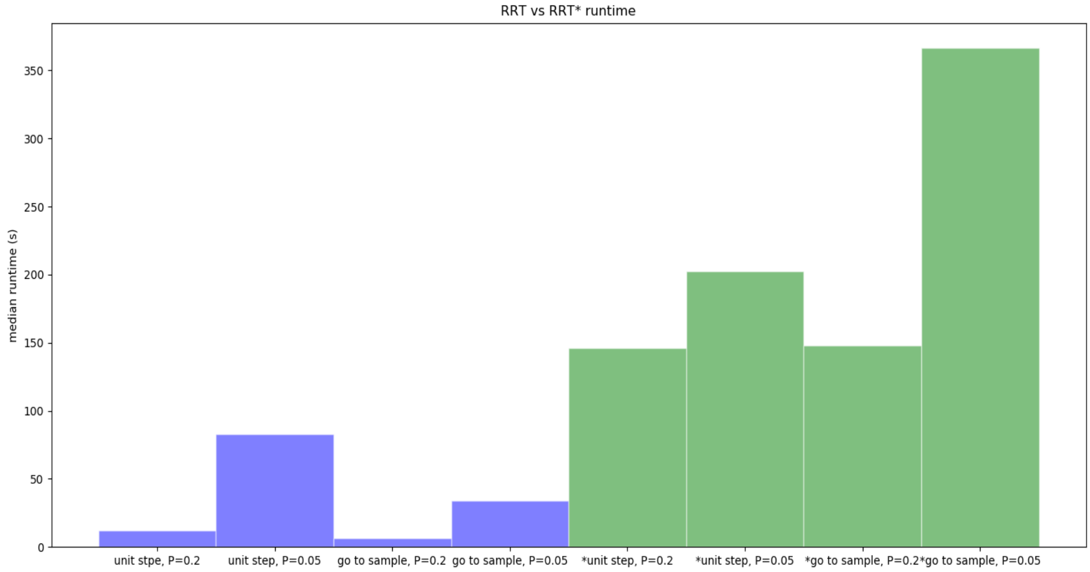


Fig 10. Runtime of RRT and RRT*. Blue bars is RRT while green bars are RRT*.

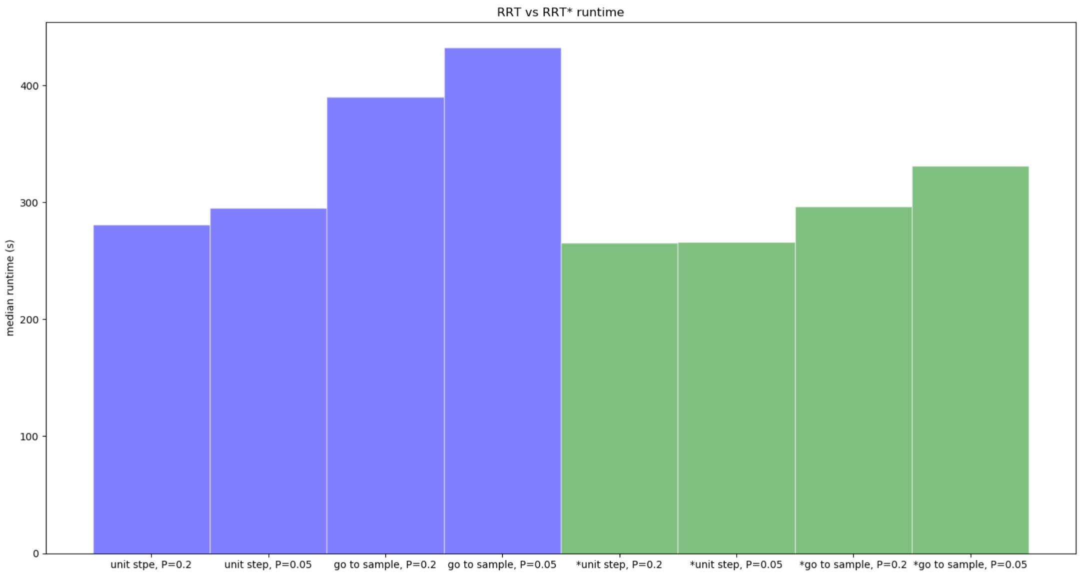


Fig 11. Path cost of RRT and RRT*. Blue bars is RRT while green bars are RRT*.

Discussion:

All my η = 1 when applying unit step.

In RRT, from Fig. 6, we can see that when applying P = 0.2, the median runtime is much less than P =0.05. And usually, extending the new point without length limitation is also faster than that only take one step size. I think it is because if one point is 200 units away, it will take 200 times for unit step to go to with 100% the same sample direction, however, for extending the new point without length limitation is just 1 time step, so it takes less time for it to extend new area.

But when comparing the path cost from Fig 7., you will notice that extending to the sampled point directly also cost higher than that of 1 step size. I think this is because the way it extends is more focused on how to reach the goal directly, if there is a chance, he will go directly, thus there would be more sharp turn along the path and thus increase the path cost.

In RRT*, from Fig 8., when P=0.05, the median runtime is much more than P=0.2. I think this is because it sampled less points on goal point, so it takes more effort on extending on overall field rather than just goal point. And also when applying unit step to extend, it takes less time on searching the goal. I think it is because in this graph, the major time consumption is on the narrow aisle, when we use unit step size, it can more easily to pass it step by step. However, if I chose to extend it to the sampled point directly, most of time, it will collide with the edge in the narrow aisle, which makes it take more effort on try and error without any progress.

When it comes to path cost in Fig 9., I found that the cost of taking unit step is less than extending the new point without length limitation. I think it is because in extending the new point without length limitation, there are less nodes inside the path which makes it has more sharp turns, thus cannot cut short the road and the path cost increases.

Conclusion:

After comparing RRT and RRT* in Fig 10. and Fig 11., generally RRT is much faster than RRT* because RRT* need to re-write the path which spending more time on it. And in RRT, I will choose η =0.2 with 1 step size, because in my experiments, it results in the shortest path and the runtime performance is also in top 2.

2. Trajectory Optimization

2.1 Representation

1.

To represent trajectories, I can use continuous time or discrete time representation

2. The $U_{smooth} = \frac{1}{2}\|\xi'(t)\|^2$

In continuous way : $U_{smooth} = \frac{1}{2}\left\|\frac{\xi(t+\Delta t)-\xi(t)}{\Delta t}\right\|^2$

In discrete way : $U_{smooth} = \frac{1}{2}\|\xi_{t+1}-\xi_t\|^2$

2.2. Inner product and gradient descent

$T_1 = (0,0,0,0,0,0)$

$T_2 = (0,5,0,0,5,0)$

$T_3 = (0,5,10,10,5,0)$

① $T_2 - T_1 = [0, 5, 0, 0, 5, 0]^T$ Euclidean distance norm $= \langle T_2 - T_1, T_2 - T_1 \rangle$

$= [0,5,0,0,5,0]\begin{bmatrix}0\\5\\0\\0\\5\\0\end{bmatrix} = 50$

$T_3 - T_1 = [0, 5, 10, 10, 5, 0]^T$ $\langle T_3 - T_1, T_3 - T_1 \rangle = [0,5,10,10,5,0]\begin{bmatrix}0\\5\\10\\10\\5\\0\end{bmatrix} = 250$

$\because \langle T_3 - T_1, T_3 - T_1 \rangle > \langle T_2 - T_1, T_2 - T_1 \rangle$

$\therefore$ T_2 trajectory is closer to T_1 trajectory

2.2.2

Hessian of $\left(\nabla_{\xi} U_{smooth}\right) = J\left(\begin{bmatrix} (q_1-q_2) \\ (q_2-q_1)-(q_3-q_2) \\ (q_3-q_2)-(q_4-q_3) \\ (q_4-q_3)-(q_5-q_4) \\ (q_5-q_4)-(q_6-q_5) \\ (q_6-q_5) \end{bmatrix}\right) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$

Thus

$$\Delta T_{21}^T H \Delta T_{21} = [0,5,0,0,5,0]\begin{bmatrix} 1 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ 0 & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 5 \\ 0 \\ 0 \\ 5 \\ 0 \end{bmatrix} = 100$$

$$\Delta T_{32}^T H \Delta T_{32} = [0,5,10,10,5,0]\begin{bmatrix} 1 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ 0 & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 5 \\ 10 \\ 10 \\ 5 \\ 0 \end{bmatrix} = 100$$

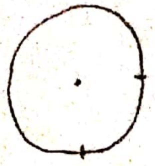
Thus, T_3 and T_2 are equally closed to T_1 #

But for $U[\xi] = \lambda U_{smooth}[\xi] + U_{close}[\xi]$

if $\lambda > 1$ (smoothness in deformation is weighted higher than closeness in position)

then T_3 will closer to T_1

2.2.3

In euclidean space (metric)  ← equall distance contour

(2D)

However in Hessian it may be 

(2D)

I still have the euclidean encoded in it because there are still $\xi^T \xi$ inside. And the Hessian metrix already consider smoothness, so, for Hessian it solves the problem.

But for euclidean metric, there no part of it encode the closeness in smoothness. So euclidean metric doesn't solve it.

3.1 Blocks and Triangles World

◦ Symbols: B0, B1, B2, B3, B4, T0, T1, Table

◦ Initial Condition: On(B0,B1), On(B1,B4), On(B2,Table), On(B3,B2), On(B4,Table),
On(T0,B0), On(T1,B3), Block(B0), Triangle(T0), Triangle(T1),
Block(B1), Block(B2), Block(B3), Block(B4)

◦ Goal Condition: On(B0,B1), On(B1,B3), On(T1,B0)

◦ Actions:

MoveBoxToTable(b1, b2)

Preconditions: On(b1, b2), Block(b1), Block(b2), Clear(b2), b1!=b2

Effects: On(b1,Table), Clear(b2), !On(b1,b2)

MoveBlockToBlock(b1,b2,b3)

Preconditions: On(b1,b2), Clear(b3), Clear(b1), Block(b1), Block(b2),
Block(b3), !Clear(b2), b1 != b2 , b2!=b3, b1!=b3

Effects: On(b1,b3), Clear(b2), !Clear(b3), !On(b1,b2)

MoveTriangleToBlock(t1,b1,b2)

Preconditions: On(t1,b1), Clear(b2), Block(b1), Block(b2), b1!=b2
Triangle(t1), !Clear(b1)

Effects: On(t1,b2), Clear(b1), !Clear(b2), !On(t1,b1)

MoveTriangleToTable(t1,b1)

Preconditions: On(t1,b1), !Clear(b1), Block(b1), Triangle(t1)

Effects: On(t1,Table), Clear(b1), !On(t1,b1)

3.2 Fire Extinguisher Environment

◦ Symbols: q, m, A, B, C, D, E, F, W, i

◦ Initial Condition: flyingAt(q, B), HighBettery(q),
EmptyTank(q), On(m, A), Buring3(F), Clear(m), Burning(F)
Location(A), Robot(m), Quadcopter(q), Location(B), Location(C),
Location(D), Location(E), Location(F), Location(W),

◦ Goal Condition: !Buring(F)

◦ Actions:

MoveToQuadcopter(q, m, l1,12)

Preconditions: flyingAt(q, l1), At(m, l2), Clear(m), l1 != l2, Location(l1), Location(L2) , Robot(m), Quadcopter(q)

Effects: At(m,l1), Robot(m), Quadcopter(q)

Landing(q, m, l)

Preconditions: flyingAt(q, l), At(m, l), Clear(m), Location(l), Robot(m), Quadcopter(q)

Effects: On(q, m), !flyingAt(q, l), !Clear(m)

Charging(q, m, l)

Preconditions: LowBettery(q), On(q,m), !Clear(m), At(m,l), Location(l), Robot(m), Quadcopter(q)

Effects: HighBettery(q)

TakingOff(q, m, l)

Preconditions: HighBettery(q), On(q,m), On(m,l), !Clear(m), Location(l), Robot(m), Quadcopter(q)

Effects: flyingAt(q, l), Clear(m),

FillingTank(q, m, W)

Preconditions: EmptyTank(q, W), On(q, m), At(m, W), !Clear(m), Location(W) , Robot(m), Quadcopter(q)

Effects: FilledTank(q),

GivingARide(q, m, l1, l2)

Preconditions: On(q, m), At(m, l1), !Clear(m), Location(l1), Location(l2), l1!=l2, Robot(m), Quadcopter(q)

Effects: At(m, l2), Clear(l1), !Clear(l2)

PouringWaterOnFire1(q, F)

Preconditions: flyingAt(q, F), Burning3(F), FilledTank(q), HIghBettery(q), Burning(F), Location(F) , Quadcopter(q)

Effects: EmptyTank(q), LowBettery(q),!Burning3(F), Burning2(F)

PouringWaterOnFire2(q, F)

Preconditions: flyingAt(q, F), Burning2(F), FilledTank(q), HIghBettery(q), Burning(F), Location(F) , Robot(m), Quadcopter(q)

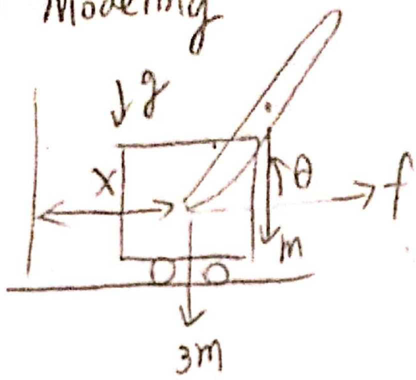
Effects: EmptyTank(q), LowBettery(q), !Burning2(F), Burning1(F)

PouringWaterOnFire3(q, F)

Preconditions: flyingAt(q, F), Burning1(F), FilledTank(q), HIghBettery(q), Burning(F), Location(F)

Effects: EmptyTank(q), LowBettery(q), !Buring1(F), !Burning(F)

4.1 Modeling



Lagrangian : $\mathcal{L} = T - V$

$= T_1 + T_2 + T_2' - V$

coordinate of cart : $\begin{bmatrix} x \\ 0 \end{bmatrix}$

coordinate of rod : $\begin{bmatrix} x + \frac{1}{2} l\cos\theta \\ \frac{1}{2} l\sin\theta \end{bmatrix}$

$T_1 = \frac{1}{2} 3m (\dot{x})^2$

$T_2 = \frac{1}{2} m \left[(\dot{\overline{x+\frac{1}{2} l\cos\theta}})^2 + (\dot{\overline{\frac{1}{2} l\sin\theta}})^2 \right]$

$V = mg \cdot \frac{l}{2}\sin\theta$

$= \frac{1}{2} m \left\{ \left[\dot{x} - \frac{1}{2} l\sin\theta\dot{\theta}\right]^2 + \left[\frac{1}{2} l\cos\theta\dot{\theta}\right]^2 \right\}$

$= \frac{1}{2} m \left\{ \dot{x}^2 - \dot{x} l\sin\theta\dot{\theta} + \frac{1}{4} l^2 \dot{\theta}^2 \right\}$

$T_2' = \frac{1}{2} I \dot{\theta}^2$

$= \frac{1}{2} \frac{ml^2}{12} \dot{\theta}^2 = \frac{1}{24} ml^2 \dot{\theta}^2$

$\mathcal{L} = T - V \quad = T_1 + T_2 + T_2' - V$

$= \frac{1}{2}(3m)\dot{x}^2 + \frac{1}{2} m\dot{x}^2 - \frac{1}{2} m\dot{x} l\sin\theta\dot{\theta} + \frac{1}{8} ml^2\dot{\theta}^2 + \frac{1}{24} ml^2\dot{\theta}^2 - \frac{1}{2} mgl\sin\theta$

$= 2m\dot{x}^2 - \frac{1}{2} m\dot{x} l\sin\theta\dot{\theta} + \frac{1}{6} ml^2\dot{\theta}^2 - \frac{1}{2} mgl\sin\theta$

$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} - f = 0$

$\Rightarrow \frac{d}{dt}\left[4m\dot{x} - \frac{1}{2} ml\sin\theta\dot{\theta}\right] - 0 - f = 0$

$\Rightarrow 4m\ddot{x} - \frac{1}{2} ml\cos\theta\dot{\theta}^2 - \frac{1}{2} ml\sin\theta\ddot{\theta} - f = 0$

$$l = 2m\dot{x}^2 - \frac{1}{2}m\dot{x}l\sin\theta\dot{\theta} + \frac{1}{6}ml^2\dot{\theta}^2 - \frac{1}{2}mgl\sin\theta$$

$$\frac{d}{dt}\frac{\partial l}{\partial\dot{\theta}} - \frac{\partial l}{\partial\theta} = 0$$

$$\Rightarrow \frac{d}{dt}\left[-\frac{1}{2}m\dot{x}l\sin\theta + \frac{1}{3}ml^2\dot{\theta}\right] - \left(\frac{-1}{2}m\dot{x}l\cos\theta\dot{\theta} - \frac{1}{2}mgl\cos\theta\right)$$

$$\Rightarrow -\frac{1}{2}m\ddot{x}l\sin\theta - \frac{1}{2}m\dot{x}l\cos\theta\,\dot{\theta} + \frac{1}{3}ml^2\ddot{\theta} + \frac{1}{2}m\dot{x}l\cos\theta\,\dot{\theta} + \frac{1}{2}mgl\cos\theta = 0$$

$$\Rightarrow -\frac{1}{2}m\ddot{x}l\sin\theta + \frac{1}{3}ml^2\ddot{\theta} + \frac{1}{2}mgl\cos\theta = 0$$

$$\Rightarrow -3\ddot{x}\sin\theta + 2l\ddot{\theta} + 3g\cos\theta = 0$$

Results: $\begin{cases} 4m\ddot{x} - \frac{1}{2}ml\cos\theta\dot{\theta}^2 - \frac{1}{2}ml\sin\theta\ddot{\theta} - f = 0 \\ -3\ddot{x}\sin\theta + 2l\ddot{\theta} + 3g\cos\theta = 0 \end{cases}$

$$\begin{cases} 24m\ddot{x}\sin\theta - 3ml\sin\theta\cos\theta\dot{\theta}^2 - 3ml\sin^2\theta\ddot{\theta} - 6f\sin\theta = 0 \\ -24m\ddot{x}\sin\theta + 16ml\ddot{\theta} + 24gm\cos\theta = 0 \end{cases}$$

$$(16ml - 3ml\sin^2\theta)\ddot{\theta} = 3ml\sin\theta\cos\theta\dot{\theta}^2 + 6f\sin\theta - 24gm\cos\theta$$

$$\ddot{\theta} = \frac{3ml\sin\theta\cos\theta\dot{\theta}^2 + 6f\sin\theta - 24gm\cos\theta}{16ml - 3ml\sin^2\theta}$$

$$\begin{cases} 4m\ddot{x} - \frac{1}{2}ml\cos\theta\,\dot{\theta}^2 - \frac{1}{2}ml\sin\theta\,\ddot{\theta} - f = 0 \\ -3\ddot{x}\sin\theta + 2l\ddot{\theta} + 3g\cos\theta = 0 \end{cases}$$

$$\begin{cases} 16m\ddot{x} - 2ml\cos\theta\,\dot{\theta}^2 - 2ml\sin\theta\,\ddot{\theta} - 4f = 0 \\ -3m\sin^2\theta\,\ddot{x} + 2ml\sin\theta\,\ddot{\theta} + 3gm\sin\theta\cos\theta \end{cases}$$

$$(16ml - 3m\sin^2\theta)\ddot{x} = 4f + 2ml\cos\theta\,\dot{\theta}^2 - 3mg\sin\theta\cos\theta$$

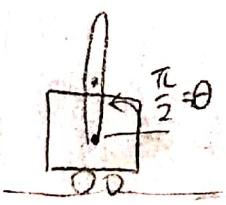
$$\ddot{x} = \frac{4f + 2ml\cos\theta\,\dot{\theta}^2 - 3mg\sin\theta\cos\theta}{16m - 3m\sin^2\theta}$$

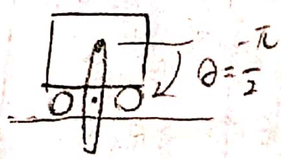
state $\underline{X} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \Rightarrow \dot{\underline{X}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dfrac{4f + 2ml\cos\theta\,\dot{\theta}^2 - 3mg\sin\theta\cos\theta}{16m - 3m\sin^2\theta} \\ \dot{\theta} \\ \dfrac{3ml\sin\theta\cos\theta\,\dot{\theta}^2 + 6f\sin\theta - 24mg\cos\theta}{16ml - 3ml\sin^2\theta} \end{bmatrix}$

Equilibrium point: $\begin{cases} \ddot{x} = 0 \\ \ddot{\theta} = 0 \\ \dot{\theta} = 0 \\ \dot{x} = 0 \\ f = 0 \end{cases} \Rightarrow \begin{cases} -3mg\sin\theta\cos\theta = 0 \\ -24mg\cos\theta = 0 \end{cases} \Rightarrow \theta = \pm\pi/2$

In $\theta = \frac{\pi}{2}$, it's a unstable equilibrium point

In $\theta = \frac{-\pi}{2}$, it's a stable equilibrium point





Scanned by CamScanner

42

$$\underline{X} = \begin{bmatrix} X \\ \theta \\ \dot{X} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ \pi/2 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} \cos\theta \approx 0 \\ \sin\theta \approx 1 \end{cases}$$

$$\dot{\underline{x}} = \begin{bmatrix} \dot{X} \\ \dot{\theta} \\ \ddot{X} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{X} \\ \dot{\theta} \\ \dfrac{4f+2ml\cos\theta\dot{\theta}^2-3mg\sin\theta\cos\theta}{16m-3m\sin^2\theta} \\ \dfrac{3ml\sin\theta\cos\theta\dot{\theta}^2+6f\sin\theta-24mg\cos\theta}{16ml-3ml\sin^2\theta} \end{bmatrix}$$

$$\dot{\underline{X}} = \begin{bmatrix} \dot{X} \\ \dot{\theta} \\ \ddot{X} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{3g}{13} & 0 & 0 \\ 0 & \frac{24g}{13l} & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \theta \\ \dot{X} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{4}{13m} \\ \frac{6}{13ml} \end{bmatrix} f = AX+Bf$$

$$y = \begin{bmatrix} X \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \theta \\ \dot{X} \\ \dot{\theta} \end{bmatrix} = CX$$

eigenvalues of matrix $A = 0, 0, \dfrac{-\sqrt{\frac{96g}{13l}}}{2}, \dfrac{\sqrt{\frac{96g}{13l}}}{2}$

Because one of my eigenvalue's real part is positive, it's unstable.

4.2 to determine if the system is controllable or not is to check the controllability matrix which is full rank or not

$$C = [B \cdot AB \quad A^2B \quad A^3B]$$

$$= \begin{bmatrix} 0 & \frac{4}{13m} & 0 & \frac{18g}{169ml} \\ 0 & \frac{6}{13ml} & 0 & \frac{144g}{169ml^2} \\ \frac{4}{13m} & 0 & \frac{18g}{169ml} & 0 \\ \frac{6}{13ml} & 0 & \frac{144g}{169ml^2} & 0 \end{bmatrix}$$

$\because$ Rank $C = 4$ $\Rightarrow$ which means the system is controllable
(from matlab

---

2. Right now it's $\begin{cases} \dot{X}(t) = AX(t) + Bu \\ y(t) = CX(t) \end{cases}$

let's design the controller to be $u = -KX(t)$

$$\Rightarrow \begin{cases} \dot{X}(t) = AX(t) + -BKX(t) \\ \quad = (A-BK)X(t) \\ y(t) = CX(t) \end{cases}$$

and $K = [k_1 \; k_2 \; k_3 \; k_4]$

$$(A-BK) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{3g}{13} & 0 & 0 \\ 0 & \frac{24g}{13l} & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \frac{4}{13m} \\ \frac{6}{13ml} \end{bmatrix} [k_1 \; k_2 \; k_3 \; k_4]$$

Let's find the eigenfunction of $(A-BK)$ $\Rightarrow \det(\lambda I-(A-BK))$

$\Rightarrow$

$$\frac{6k_2\lambda^2-6gk_1+6k_4\lambda^3-6gk_3\lambda+4k_1 l\lambda^2+4k_3 l\lambda^3+13ml\lambda^4-24mg\lambda^2}{13ml}$$

$$\Rightarrow \lambda^4+\left(\frac{4k_3 l+6k_4}{13ml}\right)\lambda^3+\left(\frac{6k_2+4k_1 l-24mg}{13ml}\right)\lambda^2+\left(\frac{-6gk_3}{13ml}\right)\lambda+\left(\frac{-6k_1 g}{13ml}\right)$$

If I want to choose my eigenvalues to be $\lambda_1, \lambda_2, \lambda_3, \lambda_4$

$(\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \lambda_4 \neq 0)$

$$(s-\lambda_1)(s-\lambda_2)(s-\lambda_3)(s-\lambda_4)$$

$$=(s^2-(\lambda_1+\lambda_2)s+\lambda_1\lambda_2)(s^2-(\lambda_3+\lambda_4)s+\lambda_3\lambda_4)$$

$$= s^4-(\lambda_1+\lambda_2)s^3+\lambda_1\lambda_2 s^2$$
$$-(\lambda_3+\lambda_4)s^3+(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)s^2-\lambda_1\lambda_2(\lambda_3+\lambda_4)s$$
$$\lambda_3\lambda_4 s^2-\lambda_3\lambda_4(\lambda_1+\lambda_2)s+\lambda_1\lambda_2\lambda_3\lambda_4$$

$$= s^4-(\lambda_1+\lambda_2+\lambda_3+\lambda_4)s^3+(\lambda_1\lambda_2+(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)+\lambda_3\lambda_4)s^2$$
$$-[\lambda_1\lambda_2(\lambda_3+\lambda_4)+\lambda_3\lambda_4(\lambda_1+\lambda_2)]s+\lambda_1\lambda_2\lambda_3\lambda_4$$

① $\lambda_1\lambda_2\lambda_3\lambda_4 = \dfrac{-6k_1 g}{13ml}$ $\Rightarrow$ $k_1 = \dfrac{(\lambda_1\lambda_2\lambda_3\lambda_4)\cdot(13ml)}{-6g}$

② $-(\lambda_1\lambda_2\lambda_3+\lambda_1\lambda_2\lambda_4+\lambda_1\lambda_3\lambda_4+\lambda_2\lambda_3\lambda_4) = \dfrac{-6g\,k_3}{13ml}$

$\Rightarrow$ $k_3 = \dfrac{(\lambda_1\lambda_2\lambda_3+\lambda_1\lambda_2\lambda_4+\lambda_1\lambda_3\lambda_4+\lambda_2\lambda_3\lambda_4)\times 13ml}{6g}$

③ $\dfrac{6k_2+4k_1 l-24mg}{13ml} = \lambda_1\lambda_2+(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)+\lambda_3\lambda_4$

$\Rightarrow$ $k_2 = \dfrac{[\lambda_1\lambda_2+(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)+\lambda_3\lambda_4]\times 13ml+24mg+\dfrac{4(\lambda_1\lambda_2\lambda_3\lambda_4)\cdot(13ml)}{6g}}{6}$

④ $\dfrac{4k_3 l+6k_4}{13ml} = -(\lambda_1+\lambda_2+\lambda_3+\lambda_4)$

$\Rightarrow$ $k_4 = \dfrac{-(\lambda_1+\lambda_2+\lambda_3+\lambda_4)\times 13ml - 4l\cdot\dfrac{(\lambda_1\lambda_2\lambda_3+\lambda_1\lambda_2\lambda_4+\lambda_1\lambda_3\lambda_4+\lambda_2\lambda_3\lambda_4)\times 13ml}{6g}}{6}$

because $\lambda_1, \lambda_2, \lambda_3, \lambda_4, m, g, l$ are all known.

thus I can design a controll $u = -K\,X(t)$

$= -[k_1\ k_2\ k_3\ k_4]\,X(t)$

Scanned by CamScanner

The way I choose my controller to be $U = -KX(t)$ is decause It's easier to implement as a feedback controller. And I can controller the systems eigenvalues (system's response) by designing different values of k_1 k_2 k_3 k_4, thus place the poles to whenever I want as long as the system is controllable.

4.2 control

3. In LQR optimal design, if my cart rod system $m=1, l=1, g=10$

then my A matrix would be

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2.3077 & 0 & 0 \\ 0 & 18.4615 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0.3077 \\ 0.4615 \end{bmatrix}$$

$u = -Kx(t)$. $\quad J = \int_0^\infty (x^T Q x + u^T R u)\, dt$

and $K = R^{-1} B^T P$

and in Ricatti equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

If I choose Q to be

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \quad R = 1 \quad \begin{bmatrix} 3.5 & -25.1 & 5.7 & -6 \\ -25.1 & 1240.5 & -82.5 & 265.4 \\ 5.7 & -82.5 & 17.6 & -19.4 \\ -6 & 265.4 & -19.4 & 67.3 \end{bmatrix}$$

then after I solve the Ricatti equation, I get P, (use Matlab)

and $K = R^{-1} B^T P = \begin{bmatrix} -1 & 97.1044 & -3.5243 & 25.1167 \end{bmatrix}$

eigenvalue of $(A - BK) = \begin{bmatrix} -7.1853 \\ -2.5750 \\ -0.3738 + 0.3313i \\ -0.3738 - 0.3313i \end{bmatrix}$ ⇒ all eigenvalues' real parts are negetive

⇒ which implies now it's a stable system

⇒ I get my LQR controller $u = -K\,x(t)$

$= \begin{bmatrix} 1 & -97.1044 & 3.5243 & -25.1167 \end{bmatrix} x(t)$

4.2.3

The way I choose Q to be $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$ is because the

states are $\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$, and in LQR, cost of state in J is

x^TQx which is $x^2+10\theta^2+\dot{x}^2+100\dot{\theta}^2$, which means I focus on θ and $\dot{\theta}$, in the cart rod system, my target is to stabalize the rod position to make it stand still, thus if there are big $\dot{\theta}$, θ then the regulator will try to fix them first especially for $\dot{\theta}$.

And why I choose all place other than diagonal to be zero, is because I want to make x^TQx to be positive definite. Because if I want to minize the cost function and there are possibility to be negative, then it will don't care about stabelizing the states, and make the x^TQx to be more negative anyway. Thus, make the x^TQx to be positive definite or semidefinite is important.

And the way I choose R to be 1, is because I think input (power, energy consumption) is relatively not important. Thus, I just set a relative small number for it. and also guarantee the u^TRu to be positive definite or semidefinite, then it's enough.