PAH Motion Planning Tutorial

Unknown Author

April 28, 2015

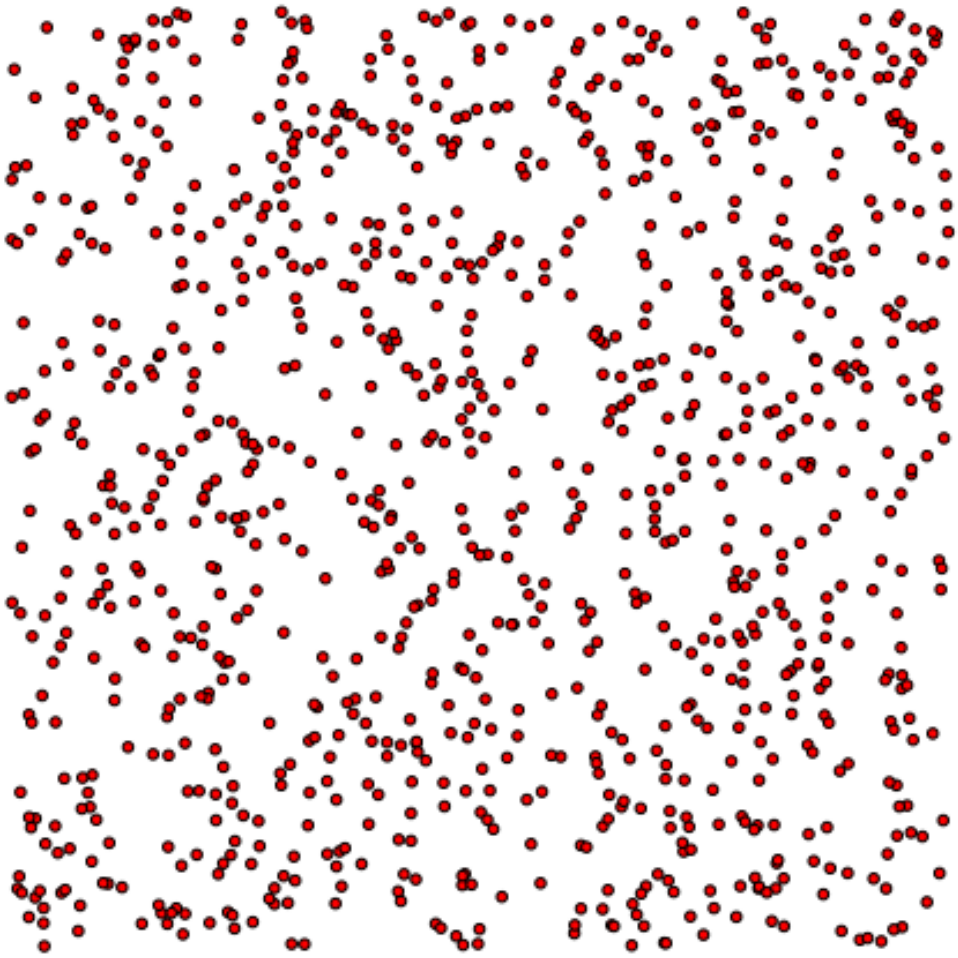
In [4]:
```
import networkx as nx
import matplotlib.pyplot as plt
import math
import numpy as np
```

1 PRM / Random r-disk graph

In [5]:
```
def rgg(n,r):
    G=nx.random_geometric_graph(n,r)
    # position is stored as node attribute data for random_geometric_graph
    pos=nx.get_node_attributes(G,'pos')

    # find node near center (0.5,0.5)
    dmin=1
    ncenter=0
    for n in pos:
        x,y=pos[n]
        d=(x-0.5)**2+(y-0.5)**2
        if d<dmin:
            ncenter=n
            dmin=d

    # color by path length from node near center
    #p=nx.single_source_shortest_path_length(G,ncenter)

    plt.figure(figsize=(8,8))
    nx.draw_networkx_edges(G,pos,nodelist=[ncenter],alpha=0.4)
    nx.draw_networkx_nodes(G,pos,nodelist=pos,
                           node_size=20)

    plt.xlim(-0.05,1.05)
    plt.ylim(-0.05,1.05)
    plt.axis('off')
    plt.show()
```

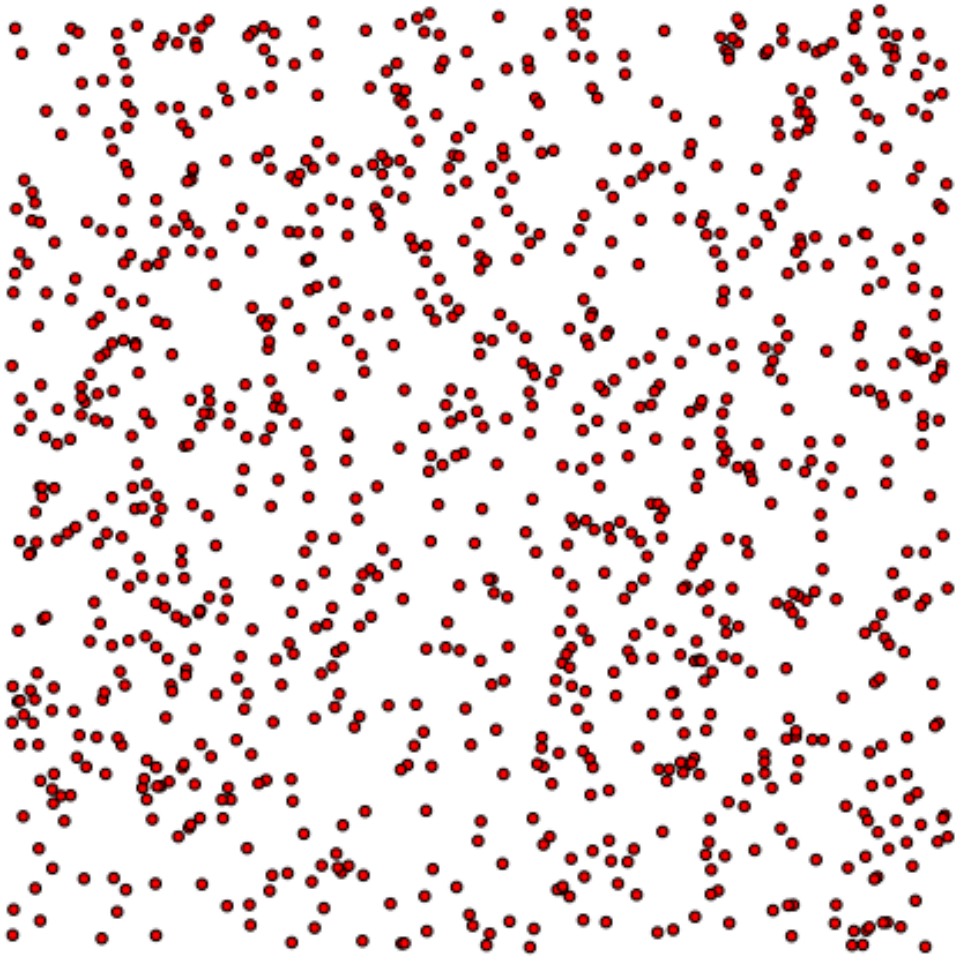
Generate 1000 uniformly distributed random points

In [6]:
```
%matplotlib inline
rgg(1000,0.0)
```



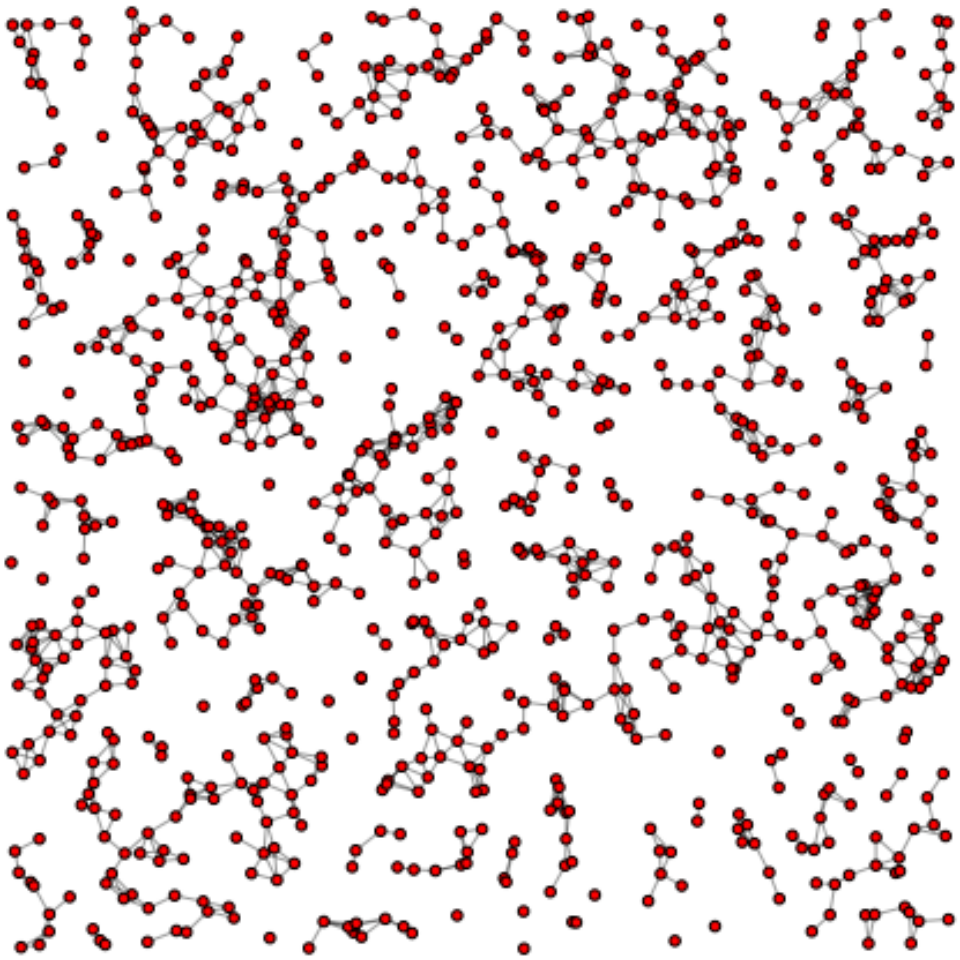
Create graph by connecting points that are at most r distant, r=0.01:

In [16]:
```
%matplotlib inline
rgg(1000,0.01)
```



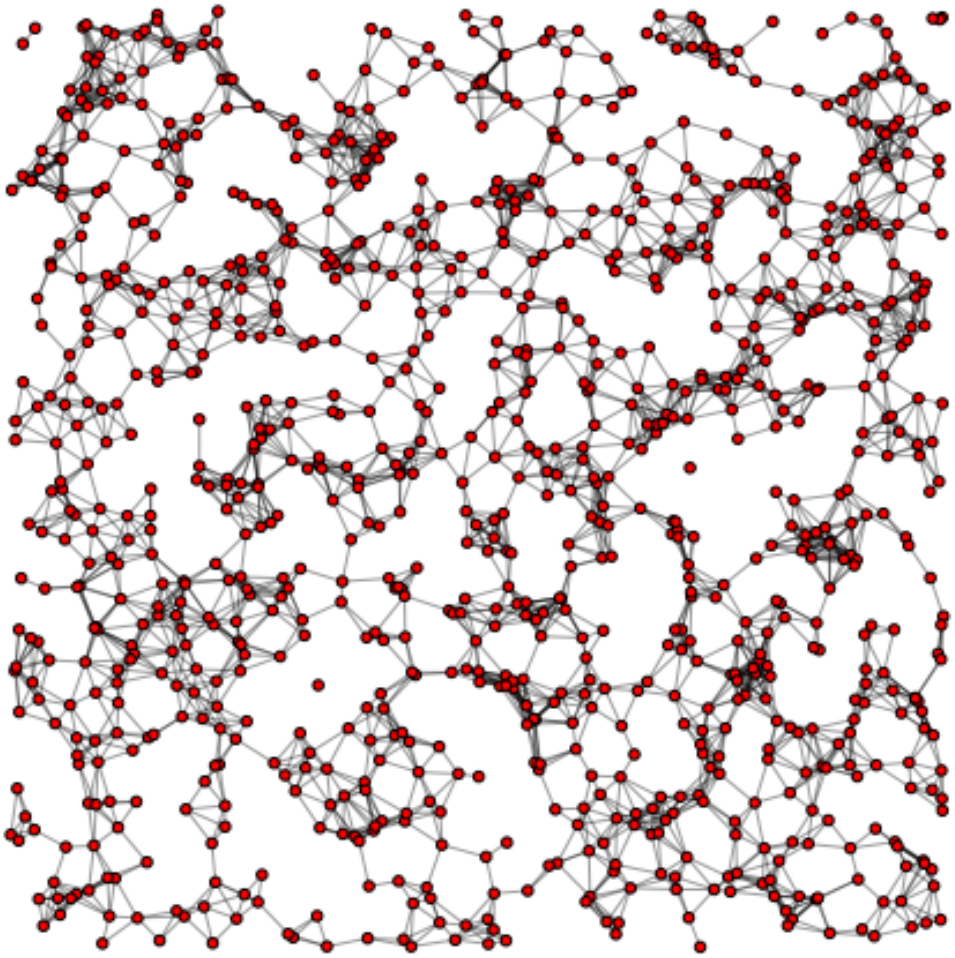
r=0.035

In [19]:
```
%matplotlib inline
rgg(1000,0.035)
```



r=0.05

In [17]:
```
%matplotlib inline
rgg(1000,0.05)
```



1.1 How to chose the radius properly?

- Percolation

$$r > (\lambda_c/n)^{1/d},$$

where $0.696 < \lambda_c < 3.372$, from simulation $\lambda_c \approx 1.44$ for $d = 2$

- Connectivity

$$r > (\frac{1}{\zeta_d}\frac{\log(n)}{n})^{1/d},$$

where $\zeta_2 = \pi$

- Asymptotic optimality

$$r = \gamma_{PRM}(\frac{\log(n)}{n})^{1/d},$$

where $\gamma_{PRM} > 2(1+1/d)^{1/d}(\mu(\mathcal{X}_{\{\nabla]]})/\zeta_d)^{1/d}$

1.2 In 2-d:

- Percolation

$$r > \sqrt{1.44/n},$$

expected number of nodes within connection radius: 4.5

- Connectivity

$$r > \sqrt{\frac{1}{\pi}\frac{\log(n)}{n}},$$

expected number of nodes within connection radius: $\log(n)$

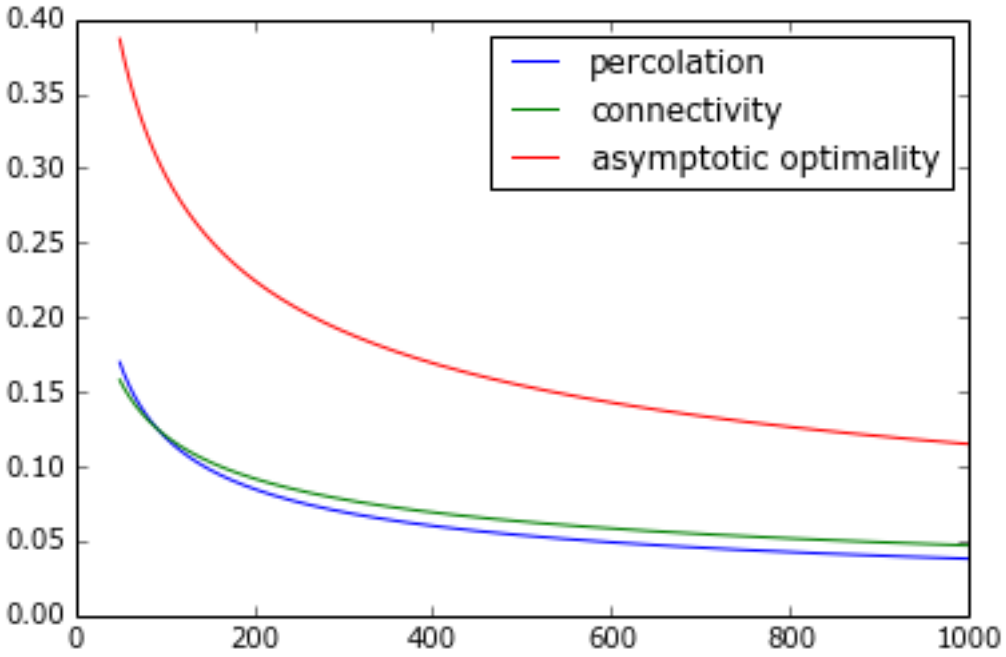
- Asymptotic optimality

$$r = \gamma_{PRM}\sqrt{\frac{\log(n)}{n}},$$

where $\gamma_{PRM} > 2\sqrt{1.5}\sqrt{\frac{1}{\pi}} \approx 1.38$ expected number of nodes within connection radius: $6\log(n)$

In [50]:
```
n = np.linspace(50,1000,1000)
r_p = np.sqrt(1.44/n)
r_c = np.sqrt((1/math.pi) * (np.log(n)/n))
gamma = 2*math.sqrt(1.5)*math.sqrt(1/math.pi)
r_o = gamma * np.sqrt(np.log(n)/n)
print("GammaPRM:", gamma)

plt.plot(n, r_p, label="percolation")
plt.plot(n, r_c, label="connectivity")
plt.plot(n, r_o, label="asymptotic optimality")
plt.legend()
plt.show()
```

```
GammaPRM: 1.3819765978853418
```


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
200
400
600
800
1000
percolation
connectivity
asymptotic optimality

In []: