

## problem\_5\_3

November 29, 2021

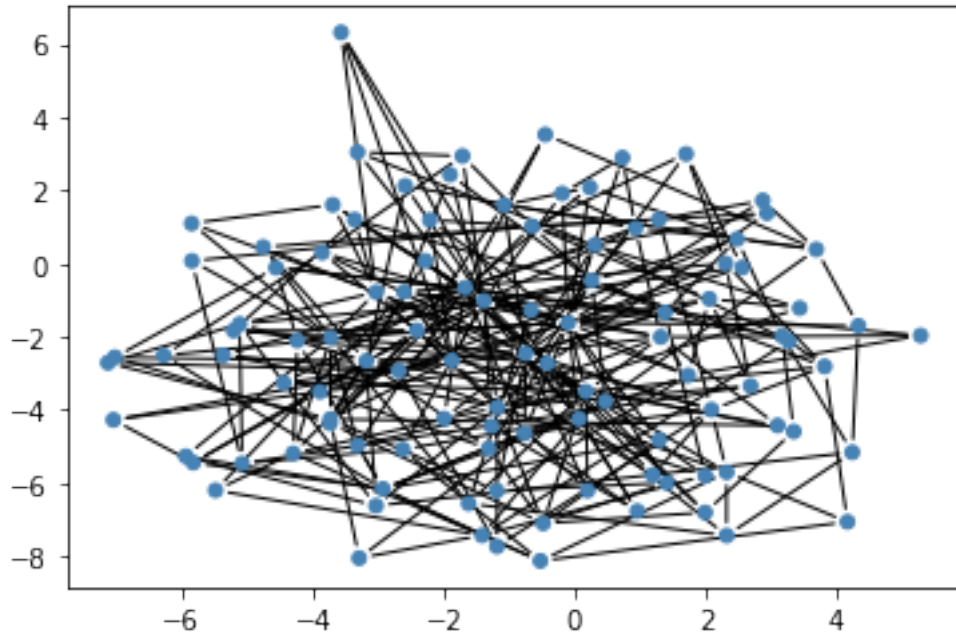
```
[1]: # Names: Klaus Kades, Lucas-Raphael Müller, Melanie Schellenberg, Shuhan Xiao
```

```
[2]: import matplotlib.pyplot as plt
import numpy as np
from igraph import *
```

```
[3]: # 5-3.1
def barabasi_albert(G, t, m):
    if m > G.vcount():
        raise "m needs to be smaller than m_0!"
    for s in range(t):
        p_ki = []
        for k in g.degree():
            p_ki.append(k / (float(G.ecount())*2))
        new_node_name = G.vcount()
        G.add_vertex(new_node_name)
        new_edges = [(new_node_name, e) for e in np.random.choice(list(range(0,
→len(p_ki))), m, replace=False, p=p_ki)]
        G.add_edges(new_edges)
```

```
[4]: # 5-3.2
m0 = 5
g = Graph.Full(m0)
barabasi_albert(g, 100, 3)
fig, ax = plt.subplots()
plot(g, layout=g.layout("kk"), target=ax)
print(f"5-3.2: a) N={g.vcount()}, b) L={g.ecount()}, c) <k>={np.sum(g.degree())/
→g.vcount()}")
```

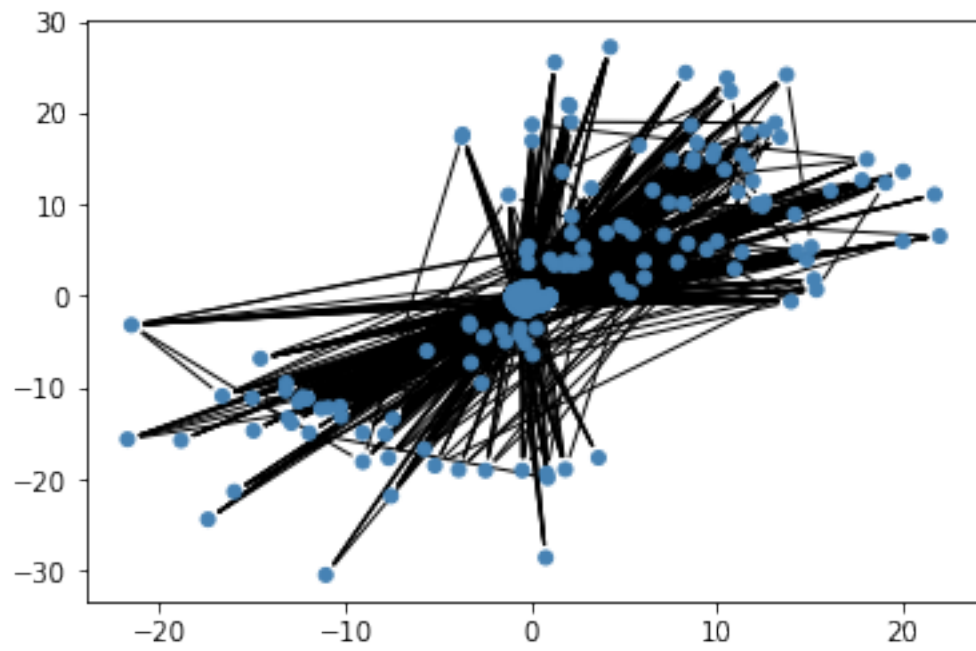
5-3.2: a) N=105, b) L=310, c) <k>=5.904761904761905



```
[5]: # 5-3.3
g = Graph.Full(5)
barabasi_albert(g, 1000, 4)
d = g.degree()
fit = statistics.power_law_fit(d, 10)
fig, ax = plt.subplots()
plot(g, layout=g.layout("kk"), target=ax)
print(f"5-3.3: \n \
    a) <C>={np.mean(g.transitivity_local_undirected())} with {np.log(g.
↪vcount())*np.log(g.vcount())/g.vcount()},\n \
    b) <d>={g.average_path_length()} with {np.log(g.vcount()) / np.log(np.log(g.
↪vcount()))}) \n \
    c) lambda={fit.alpha} with 3 for Barabási-Albert Model")
```

5-3.3:

- a)  $\langle C \rangle = 0.03148977602687381$  with  $0.04754827194256712$ ,
- b)  $\langle d \rangle = 3.2138391706804623$  with  $3.5754953042028204$
- c)  $\lambda = 2.8636287570611785$  with 3 for Barabási-Albert Model



[ ]: