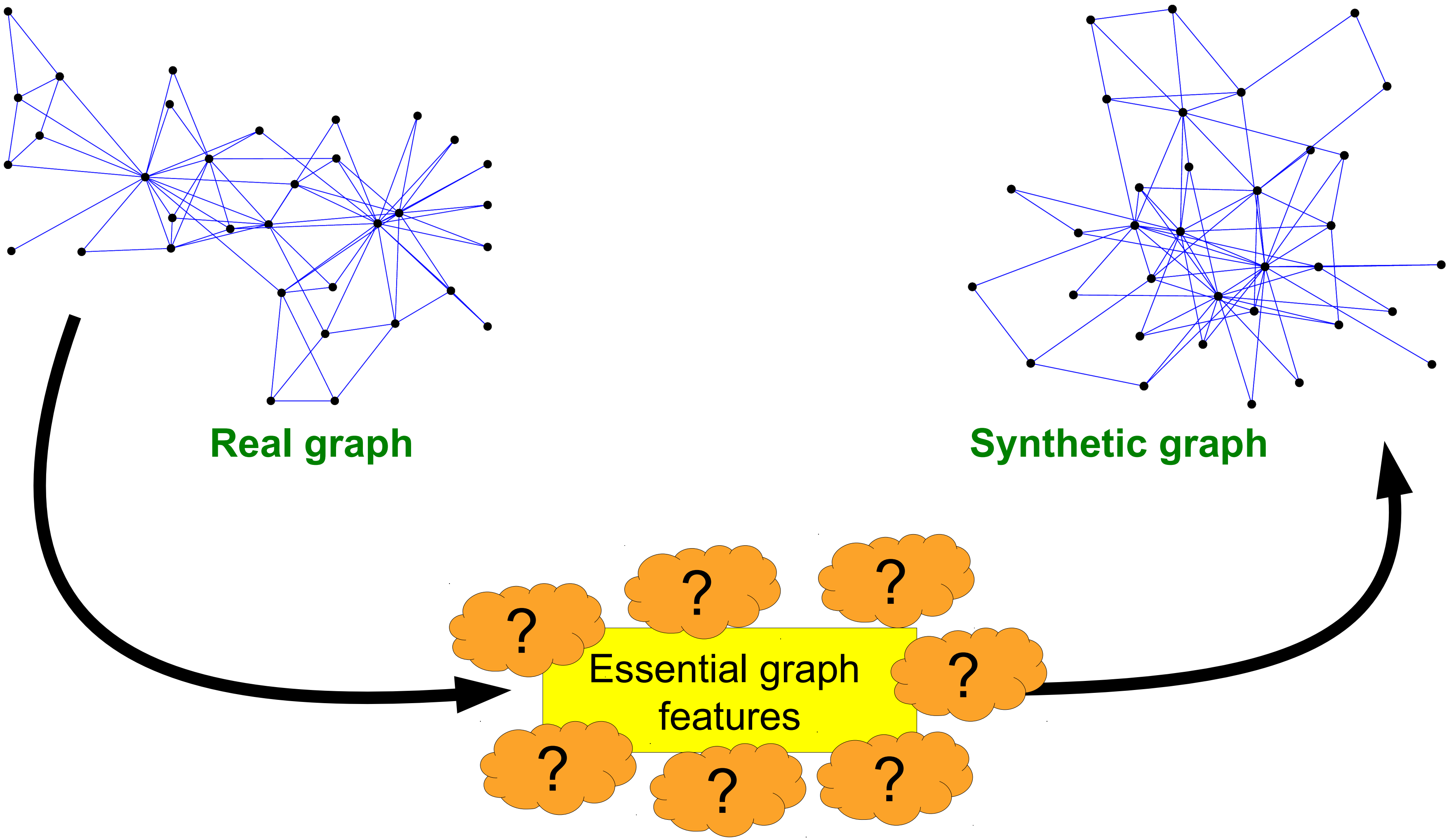
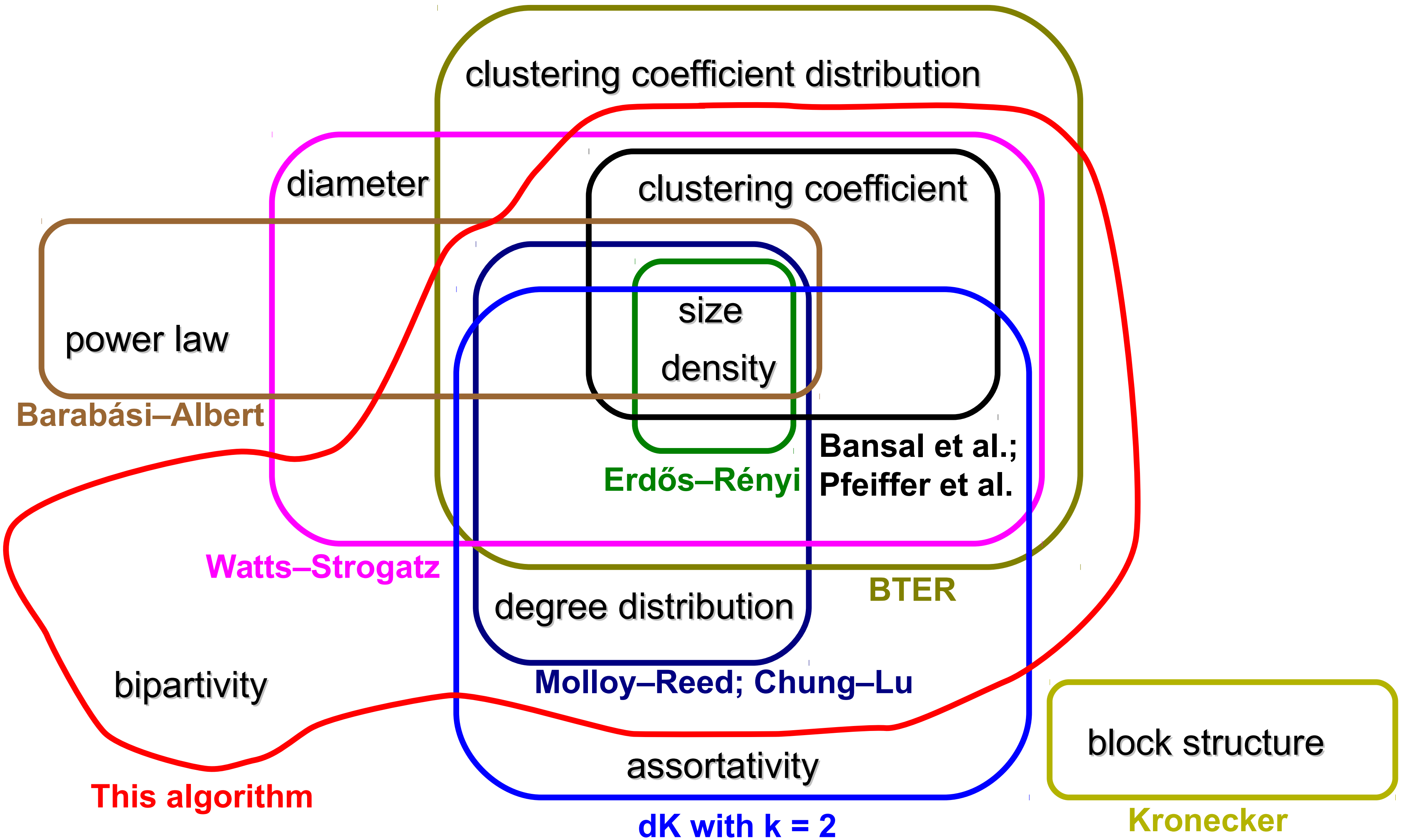
Generating Networks with Realistic Properties Based on a Given (Set of) Network(s)

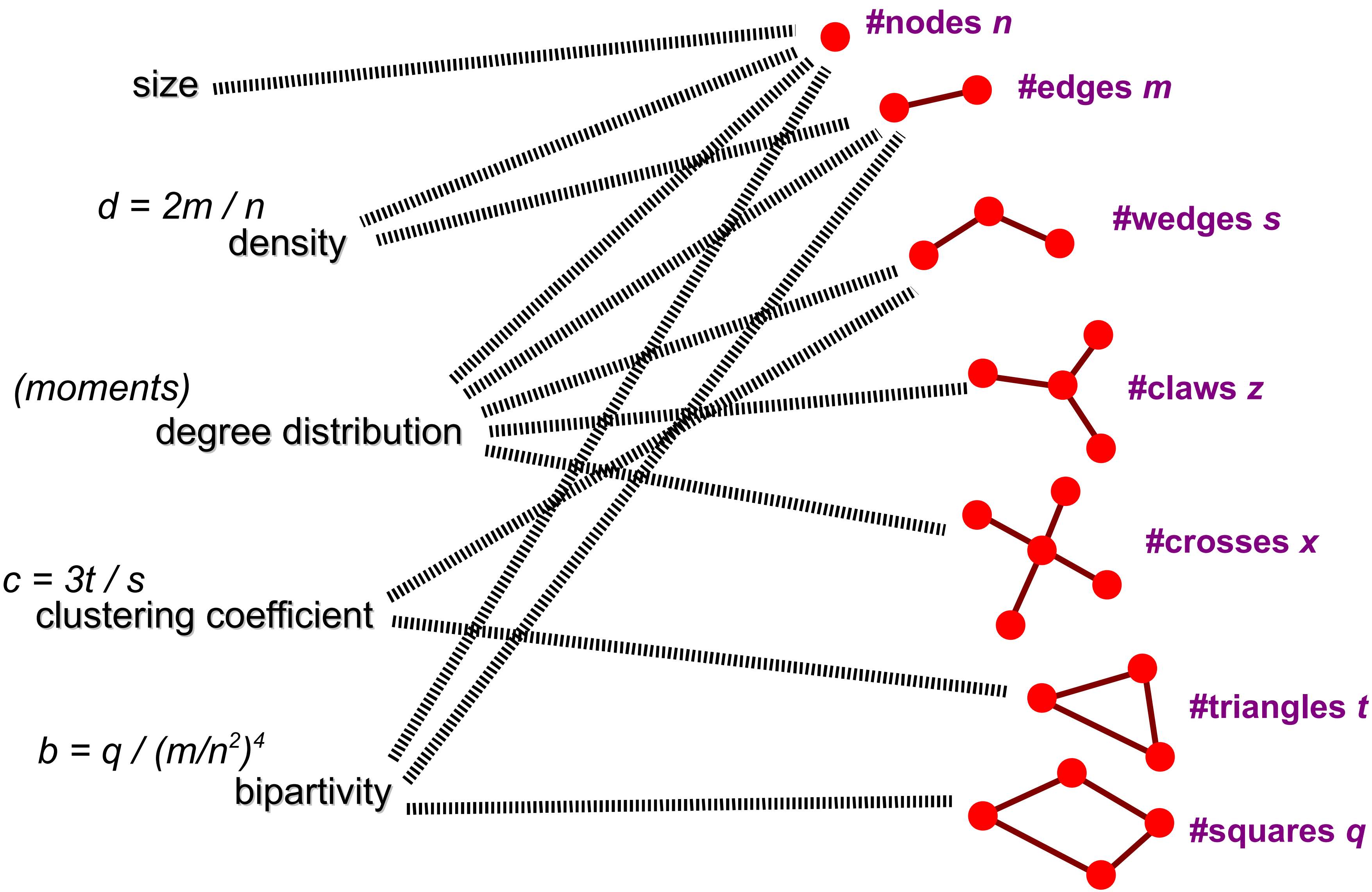
Part I: Network Models



Essential Features



Mapping Features to Statistics



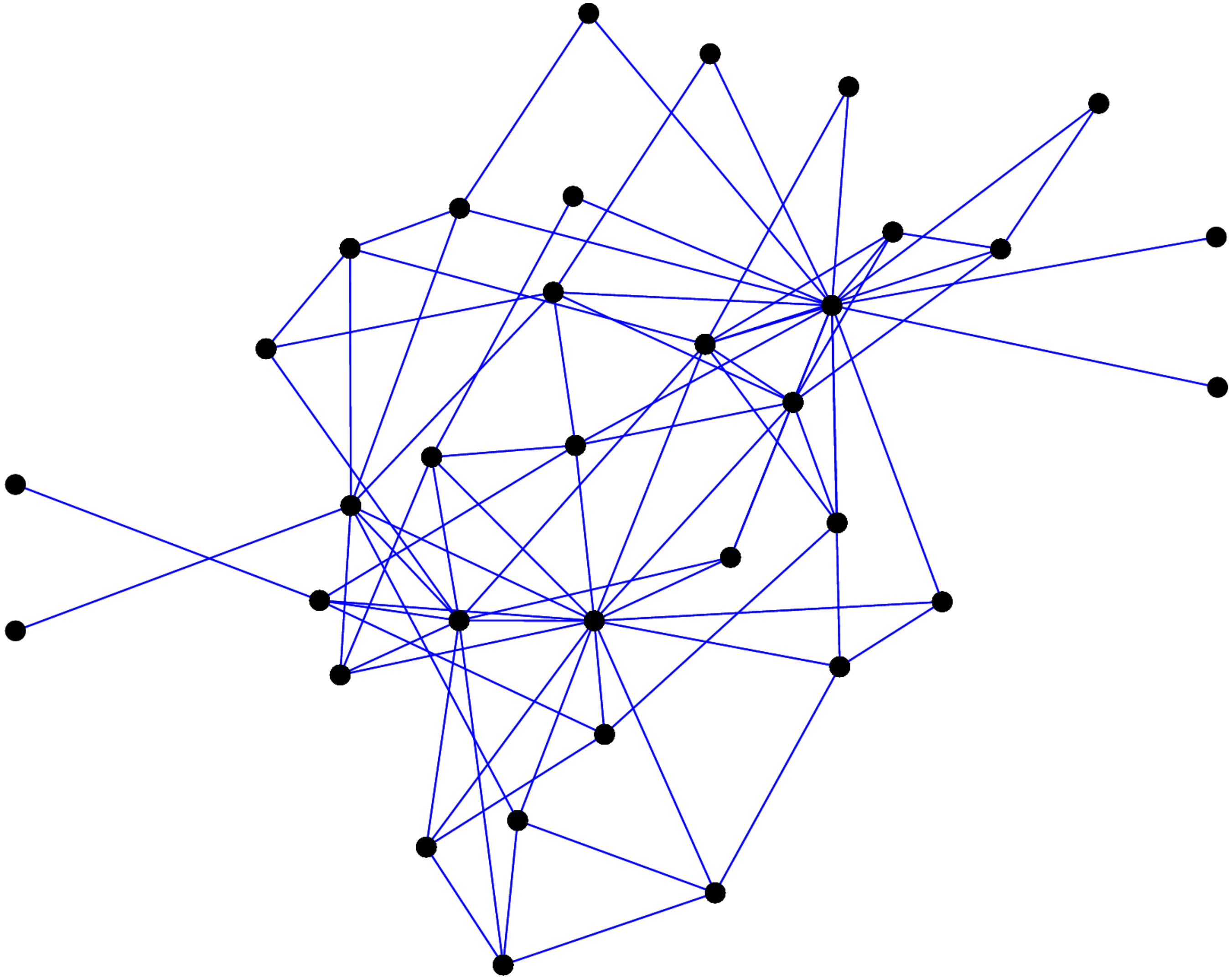
Algorithm – Vectorized Version

```
G = ErdősRényi(n, p)
for all statistics S {
    x_S = S(G)
    y_S = S(G_0)
}

repeat {
    Choose node u at random
   for all statistics S {
        Δ_S = Diffvect(G, u, S)
    }
    E = sum_S ((y_S + Δ_S − x_S) / x_S)^2
    v = argmin_w≠u E_w
    G = G ± {uv}
    for all statistics S { y_S = y_s + Δ_S_w }
} until  E has not reached a new minimum value in the
  last (−n ln ε) iterations
```

Compute changes of statistic S for flips between u and all other nodes

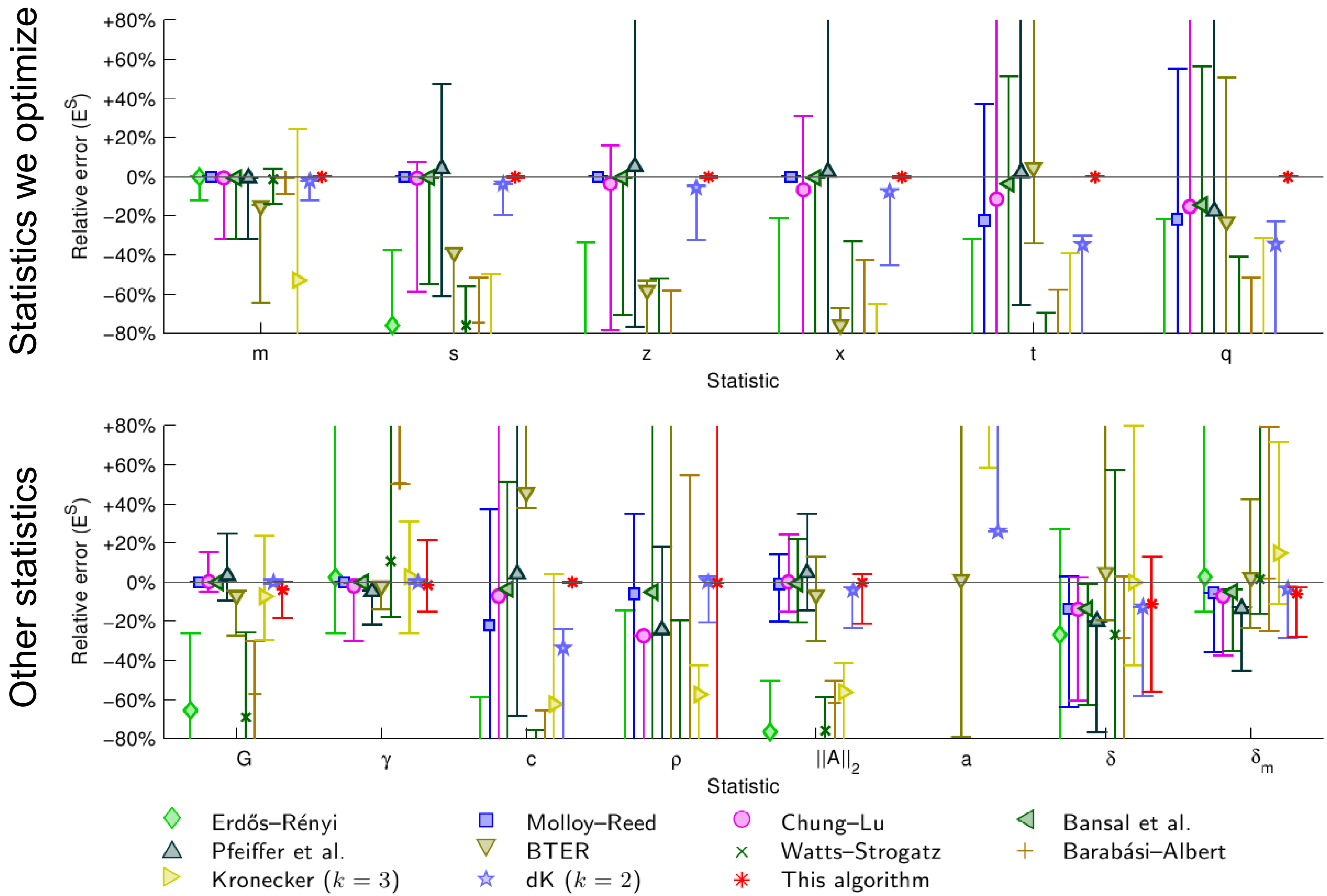
Generate Network with Same Properties as Zachary's Karate Club



http://konect.cc/networks/ucidata-zachary/

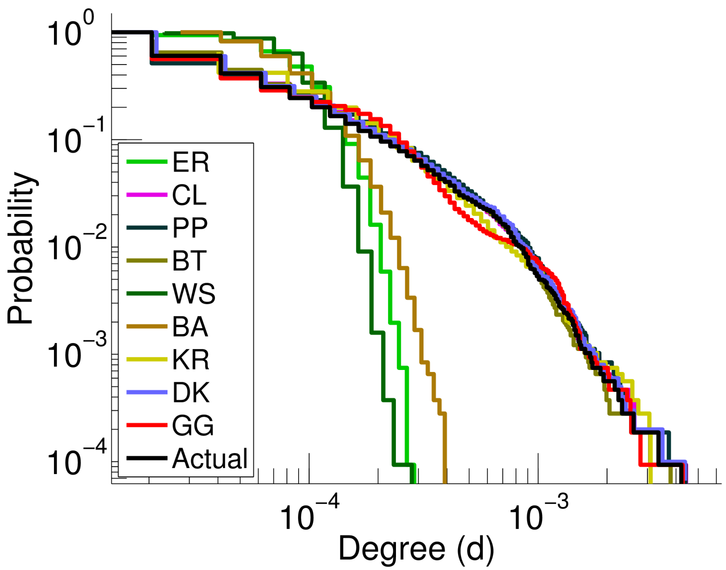
Experiment: Precision (all datasets)

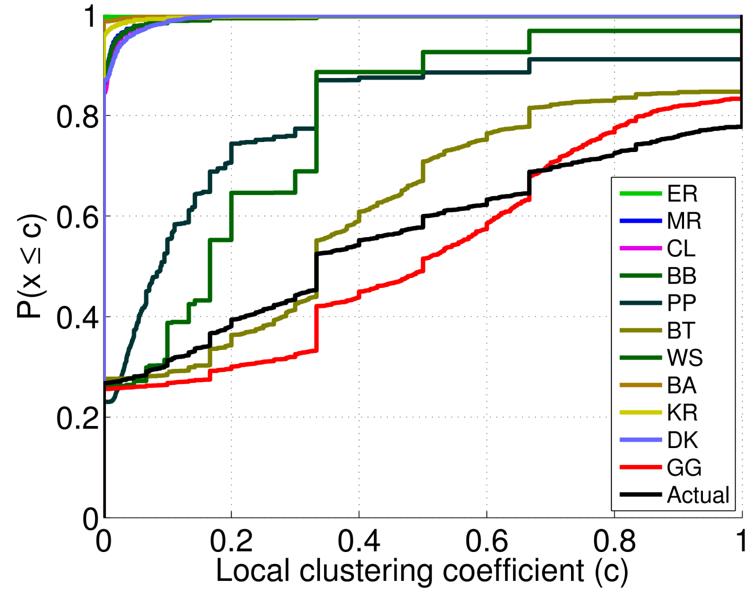
Plots show median and 10th/90th percentiles over 36 networks

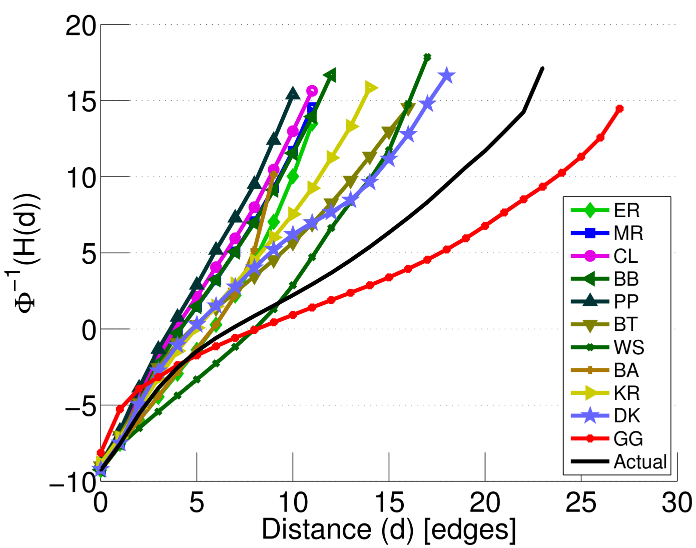


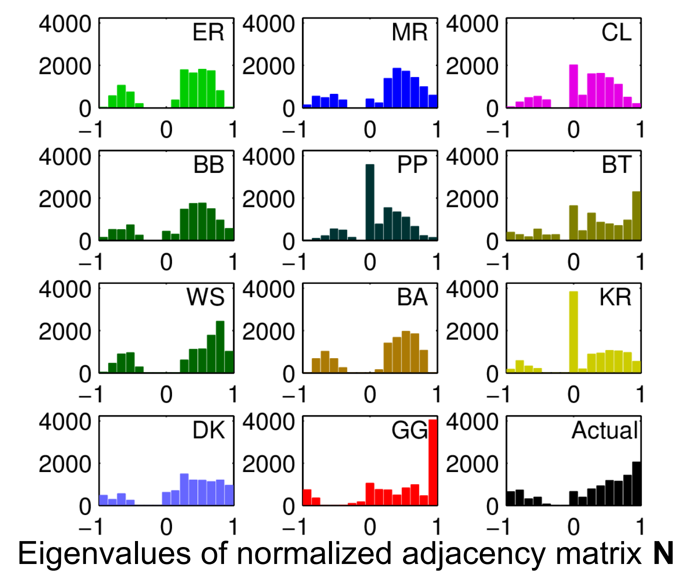
Qualitative Experiments

(Pretty Good Privacy network)

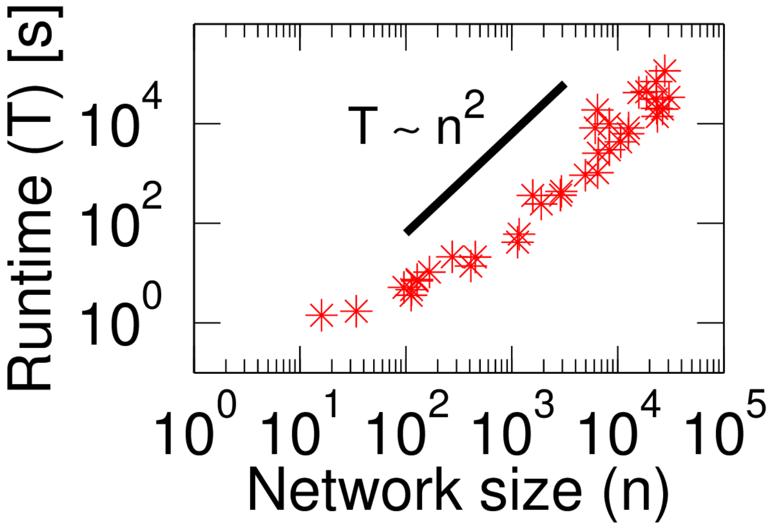


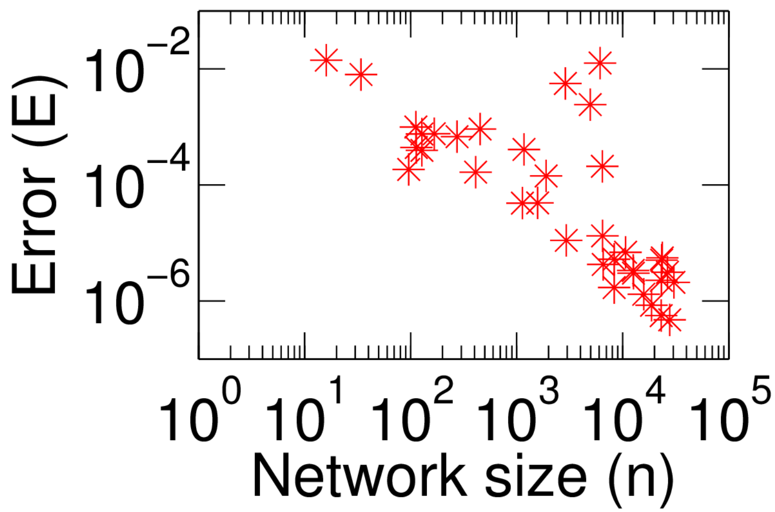




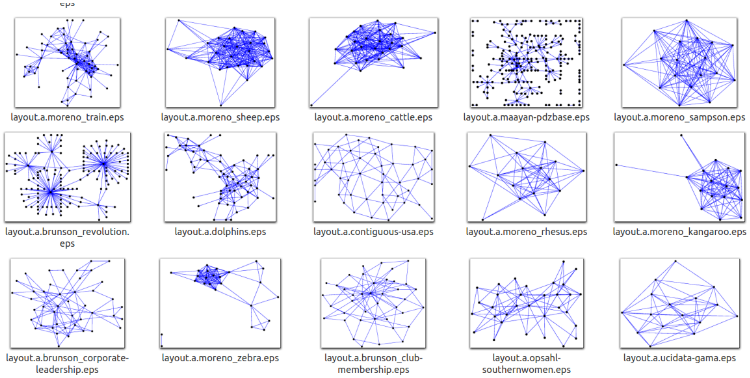


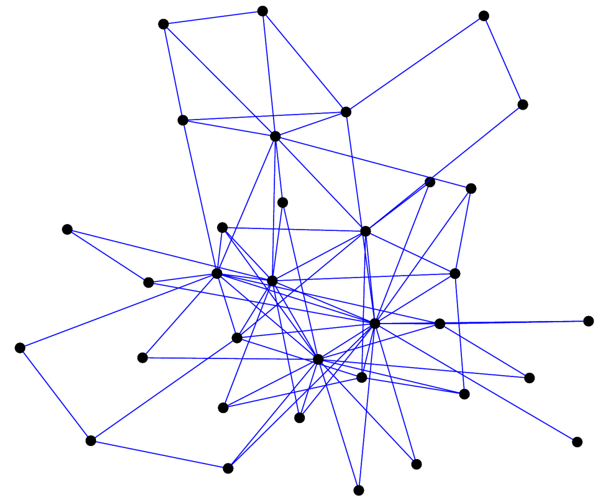
Experiment: Scalability





Part II: Network Set Models



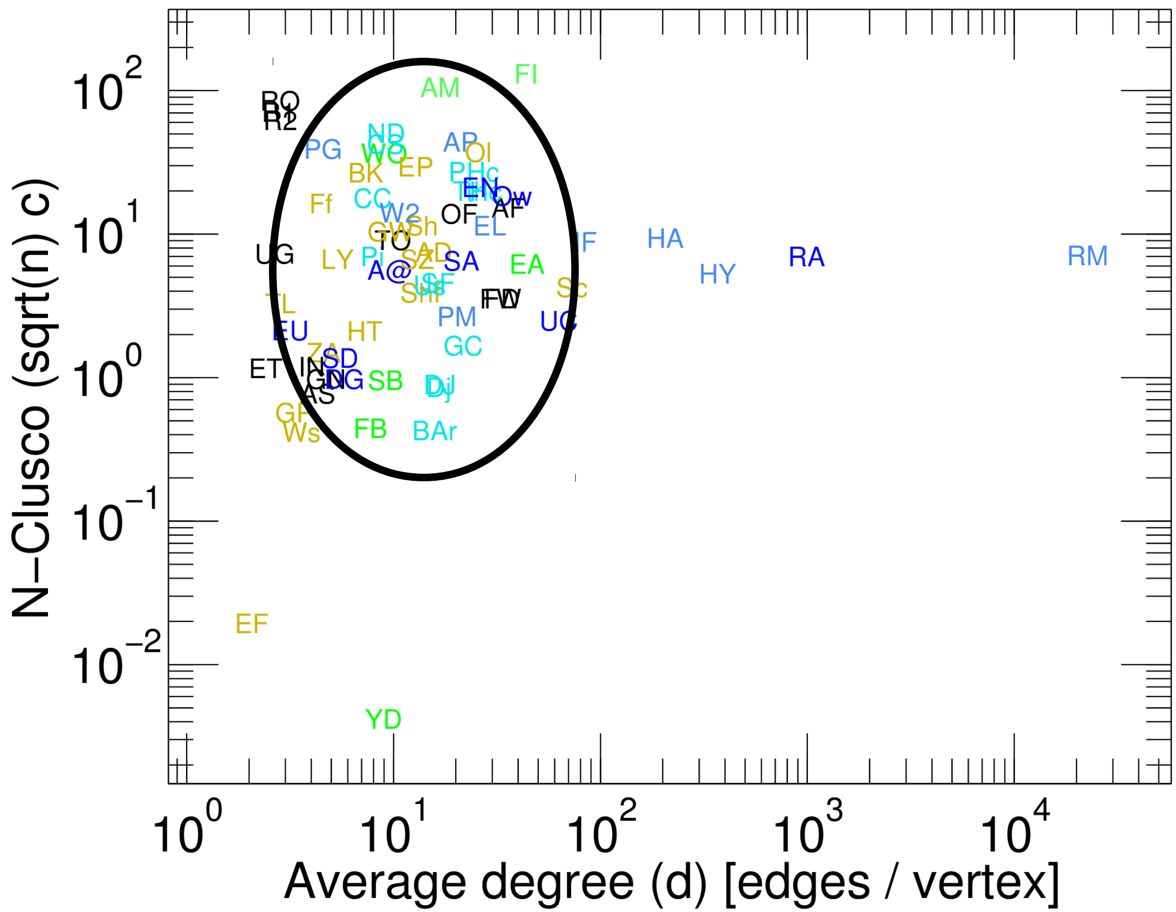


Real graphs

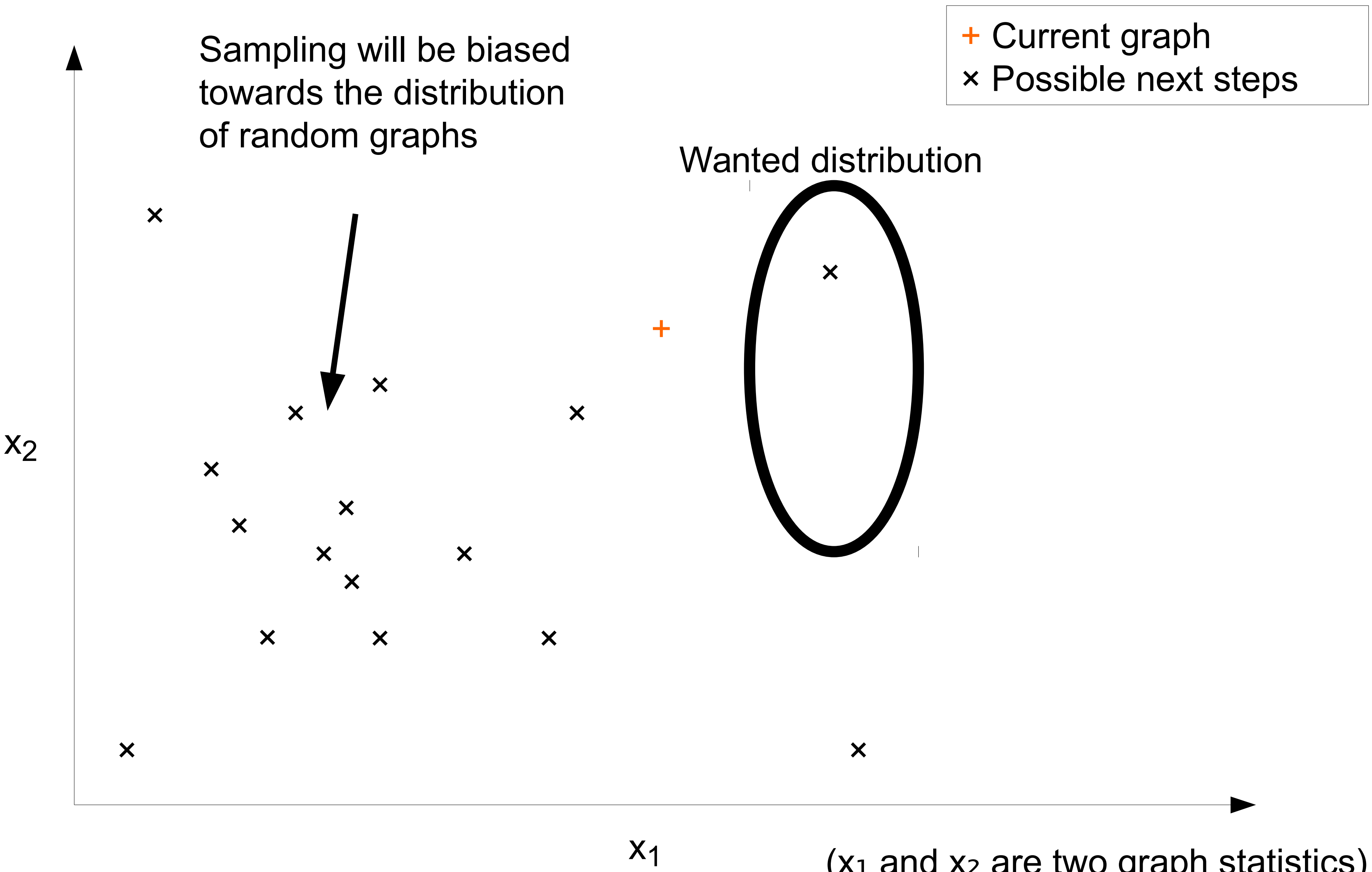
Synthetic graph

Essential graph features

Real Networks Have a Distribution of Values

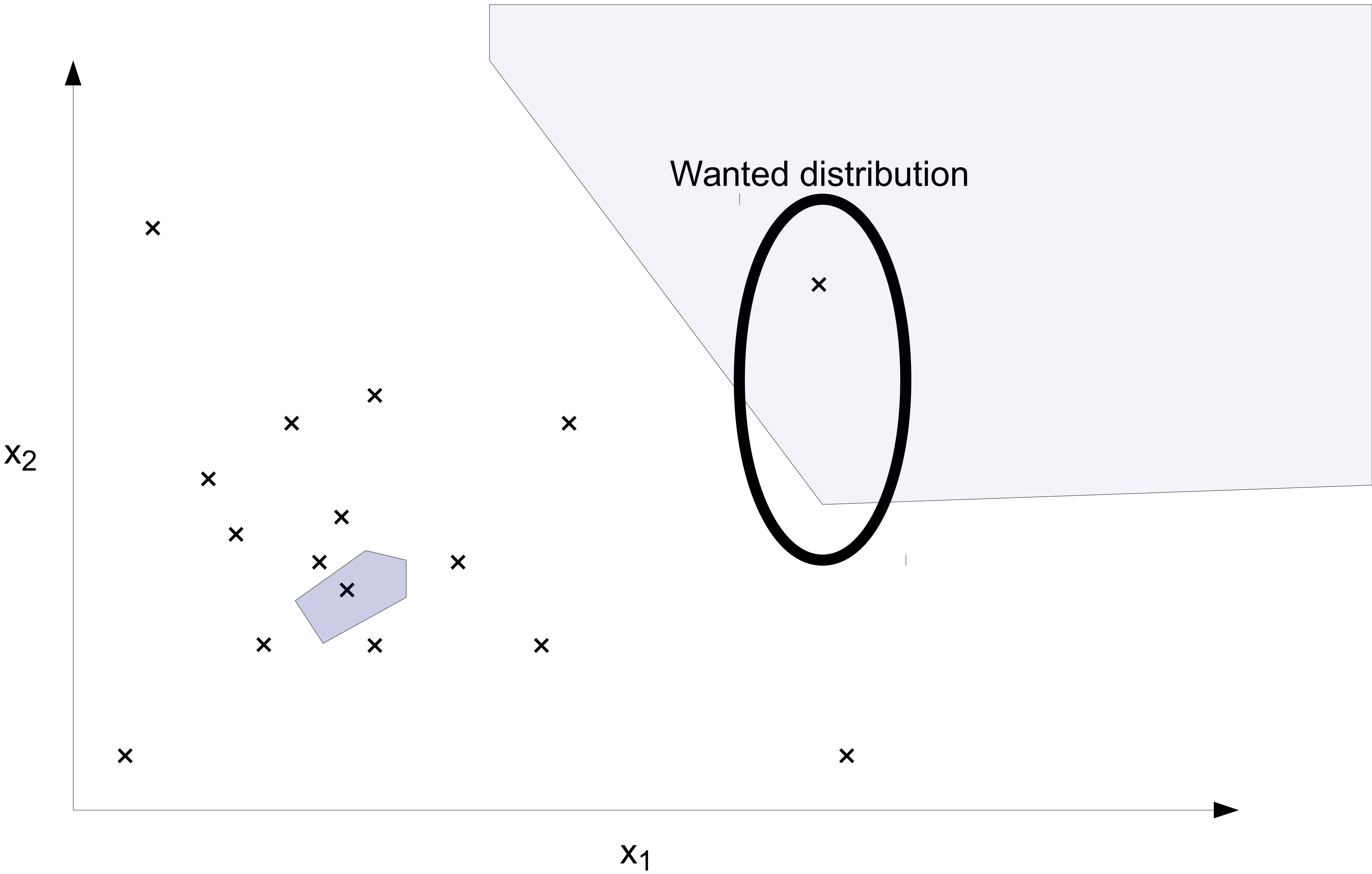


Monte Carlo Markov Chain Methods



(x1 and x2 are two graph statistics)

Solution: Integral of Measure of Voronoi Cells



How to Compute the Integral over Voronoi Cells

Answer: We don't have to.

Sampling strategy:

- Sample point in statistic-space according to our wanted distribution
- Find nearest possible network (i.e., nearest "×")

Claim: This distribution at each step is similar to the underlying measure, giving an unbiased sampling.

Result: Close, But Not Exact