

# Empirical Investigation of an Open Conjecture:

Let  $G \sim G(n, c/n)$  be an Erdős-Rényi random graph with mean degree

Agentic NL→Lean 4 Pipeline  
Job #11

April 26, 2026

## Abstract

This report documents the empirical investigation of an open mathematical conjecture that could not be formally proved or disproved in Lean 4 with Mathlib. Numerical experiments were conducted to gather evidence for or against the conjecture. The empirical verdict is: **Inconclusive**. The conjecture remains formally open.

## 1 Conjecture Statement

### Conjecture 1.

Let  $G \sim G(n, c/n)$  be an Erdős-Rényi random graph with mean degree  $c > 0$ . For  $\beta > 0$  and  $S \subseteq [n]$ , let  $A_S$  denote the adjacency matrix of the induced subgraph  $G[S]$ .

*[Bonacich centrality]*

For  $\beta < \rho(A_S)^{-1}$ , the *Bonacich centrality* of node  $i \in S$  within  $G[S]$  is

$$b_i(\beta, S) = \text{bigl}[(I - \beta A_S)^{-1}\mathbf{1}\bigr]_i = \sum_{k=0}^{\infty} \beta^k [A_S^k]_i.$$

*[end{definition}]*

*[Bonacich  $u$ -core]*

For a threshold  $u \geq 1$ , the *Bonacich  $u$ -core* of  $G$  at attenuation  $\beta$  is the largest subset  $S^*(u) \subseteq [n]$  such that

$$b_i(\beta, S^*(u)) \geq u \quad \forall i \in S^*(u).$$

Equivalently,  $S^*(u)$  is the maximal fixed point of the monotone operator  $T_u(S) = \{i \in S : b_i(\beta, S) \geq u\}$ .

*[end{definition}]*

*[Square-root singularity]*

Fix  $c > 1$  and  $0 < \beta < 1/c$ . There exists a critical threshold  $u_c = u_c(\beta, c) > 1$  and constants  $\phi_c = \phi_c(\beta, c) > 0$ ,  $C = C(\beta, c) > 0$  such that:

```

\begin{enumerate}
  \item[\textup{(i)}] For  $u < u_c$ , the Bonacich  $u$ -core is non-empty with high probability and
  \[
    \frac{|S^*(u)|}{n} \xrightarrow{\mathbb{P}} \phi^*(u) > 0 \quad (n \rightarrow \infty),
  \]
  \item[\textup{(ii)}] For  $u > u_c$ , the Bonacich  $u$ -core satisfies  $|S^*(u)| = o_{\mathbb{P}}(n)$ .
  \item[\textup{(iii)}] The order parameter satisfies
  \[
    \phi^*(u) - \phi_c \sim C \sqrt{u_c - u} \quad \text{as } u \rightarrow u_c^-,
  \]
\end{enumerate}
\end{conjecture}

```

## 2 Status

**Formal Status:** OPEN — no Lean 4 proof or disproof was found.

**Empirical Verdict:** **Inconclusive**

The pipeline attempted formal verification in Lean 4 with Mathlib but was unable to produce a compiling proof or disproof. Empirical testing was then conducted to gather numerical evidence.

## 3 Basic Empirical Testing

The following output was produced by the basic numerical experiment:

```

=== EXPERIMENT PLAN ===

Conjecture under test:
For  $G \sim G(n, c/n)$  with  $c > 1$  fixed and  $0 < \beta < 1/c$ , define the
Bonacich  $u$ -core as the maximal fixed point of
 $T_u(S) = \{ i \in S : b_i(\beta, S) \geq u \}$ ,
where  $b(\beta, S) = (I - \beta A_S)^{-1} \mathbf{1}$ .
Claim: there exist  $u_c > 1$ ,  $\phi_c > 0$ ,  $C > 0$  such that
(i) For  $u < u_c$ ,  $|S^*(u)|/n \xrightarrow{\mathbb{P}} \phi^*(u) > 0$ .
(ii) For  $u > u_c$ ,  $|S^*(u)| = o_{\mathbb{P}}(n)$ .
(iii)  $\phi^*(u) - \phi_c \sim C \sqrt{u_c - u}$  as  $u \rightarrow u_c^-$ .

Methodology (multiple angles):
(A) Random sampling of ER graphs at several sizes  $n$  in  $\{400, 800, 1600, 3200\}$ 
with multiple independent trials; compute  $\phi_n(u)$  for a grid of  $u$ .
(B) PHASE TRANSITION CURVE: plot  $\phi_n(u)$  vs  $u$ ; observe discontinuity.
(C) COUNTEREXAMPLE SEARCH for monotonicity:  $\phi(u)$  must be non-increasing.
(D) FINITE-SIZE SCALING: curves for larger  $n$  should approach a limit
and sharpen near  $u_c$  (first-order-like jump).
(E) CRITICAL POINT ESTIMATION: identify  $u_c$  from the drop of  $\phi_n(u)$ .
(F) SQUARE-ROOT FIT: joint non-linear regression  $\phi = \phi_c + C(u_c - u)^\alpha$ .

```

The conjecture predicts  $\alpha = 1/2$ .  
 (G) LOG-LOG SCALING: slope of  $\log(\phi - \phi_c)$  vs  $\log(u_c - u) \sim 0.5$ .  
 (H) SUPER-CRITICAL VANISHING:  $|S^*(u)|$  grows sublinearly for  $u > u_c$ .  
 (I) ROBUSTNESS across two  $(c, \beta)$  parameter choices.  
 (J) EDGE CASES:  $u=1$  (all non-negative Bonacich survive),  $u$  large (empty core), small graphs.  
 Bonacich vectors computed via Neumann iteration  $x \leftarrow 1 + \beta A x$  (convergent when  $\beta * \rho(A_S) < 1$ ) with direct-solve fallback.

MAIN:  $c=3.0$ ,  $\beta=0.2$ ,  $ns=[400, 800, 1600, 3200]$ ,  $trials=[4, 3, 2, 2]$ ,  $|u_{grid}|=30$

$n= 400$  trial=1/4 t= 0.6s  
 $n= 400$  trial=2/4 t= 1.4s  
 $n= 400$  trial=3/4 t= 2.7s  
 $n= 400$  trial=4/4 t= 3.5s  
 $n= 800$  trial=1/3 t= 4.3s  
 $n= 800$  trial=2/3 t= 5.4s  
 $n= 800$  trial=3/3 t= 6.5s  
 $n= 1600$  trial=1/2 t= 8.1s  
 $n= 1600$  trial=2/2 t= 9.7s  
 $n= 3200$  trial=1/2 t= 13.9s  
 $n= 3200$  trial=2/2 t= 17.4s

Monotonicity violations across all samples/trials: 0

Rough  $(u_c, \phi_c)$  from  $n=3200$ : (2.810, 0.612)

Free-alpha fit:  $\phi_c=0.1151 \pm 0.7968$   $C=0.7296 \pm 0.7650$   $u_c=3.0399 \pm 0.1700$   $\alpha=0.2603 \pm 0.2468$

Sqrt fit ( $\alpha=0.5$ ):  $\phi_c=0.6081$   $C=0.2821$   $u_c=2.7314$   $R^2=0.99602$

Log-log slope = 0.5336 (conjecture predicts 0.5),  $R = 0.9979$

[robust]  $n=400$  trial=1/3 t=17.6s  
 [robust]  $n=400$  trial=2/3 t=17.9s  
 [robust]  $n=400$  trial=3/3 t=18.1s  
 [robust]  $n=1600$  trial=1/2 t=18.6s  
 [robust]  $n=1600$  trial=2/2 t=19.3s

Robustness ( $c=4.0, \beta=0.15$ ):  $u_c \sim 2.136$   $\phi_c \sim 0.588$   $\alpha=0.255$

Edge-case tests:

$u=1$  on  $n=300$ : fraction of nodes with  $b_i \geq 1 = 1.0000$  (expected  $\sim 1$ )

$|S^*(1)|/n$  for  $n=300$ : 1.0000 (expected  $\sim 1$ )

$|S^*(100)|/n$  for  $n=300$ : 0.0000 (expected 0)

=== RESULTS SUMMARY ===

Parameters (main):  $c=3.0$ ,  $\beta=0.2$   
 Parameters (robust):  $c=4.0$ ,  $\beta=0.15$   
 n values: [400, 800, 1600, 3200]  
 Total graph samples: 16  
 Total u evaluations: 460  
 Wall-clock time: 20.7s

Monotonicity violations (expected 0): 0

Estimated  $u_c$  : 2.7314

Estimated  $\phi_c$  : 0.6081

Estimated C : 0.2821

```

R^2 of sqrt fit : 0.99602
Free-alpha fit : alpha = 0.2603 (conjecture: 0.5)
Log-log slope : 0.5336 (conjecture: 0.5)
Robustness alpha : 0.2553 (conjecture: 0.5)

Super-critical sizes |S*(u)| vs n:
  u=2.931 : ['140.8', '130.7', '0.0', '0.0'] (ratio last/first = 0.00, n
    ratio = 8.0)
  u=3.231 : ['127.0', '0.0', '0.0', '0.0'] (ratio last/first = 0.00, n
    ratio = 8.0)
  u=3.731 : ['0.0', '0.0', '0.0', '0.0'] (ratio last/first = 0.00, n ratio
    = 8.0)

Scorecard:
  phi(u) monotone non-increasing
  phi_c > 0 (=0.608)
  super-critical sub-linear (ratio 0.00 vs n ratio 8.00)
  alpha = 0.26026111500701543
  sqrt fit R^2=0.9960
  log-log slope = 0.534
  robustness alpha = 0.25530274409283027

5/7 empirical criteria satisfied.

=== VERDICT ===
INCONCLUSIVE: 5/7 criteria met. Transition and phi_c>0 confirmed but
  exponent / scaling details (=0.260, log-log slope=0.534, R^2=0.9960) de
... [truncated]

```

## 4 Experiment Code (Basic)

```

import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import scipy.sparse as sp
from scipy.sparse.linalg import spsolve
from scipy.stats import linregress
from scipy.optimize import curve_fit
import time
import warnings
warnings.filterwarnings("ignore")

print("===_EXPERIMENT_PLAN_===")
print("""
Conjecture under test:
  For  $G \sim G(n, c/n)$  with  $c > 1$  fixed and  $0 < \beta < 1/c$ , define the
  Bonacich  $u$ -core as the maximal fixed point of
     $T_u(S) = \{ i \text{ in } S : b_i(\beta, S) \geq u \}$ ,
  where  $b(\beta, S) = (I - \beta A_S)^{-1} \mathbf{1}$ .
  Claim: there exist  $u_c > 1$ ,  $\phi_c > 0$ ,  $C > 0$  such that

```

- (i) For  $u < u_c$ ,  $|S^*(u)|/n \rightarrow_P \phi^*(u) > 0$ .
- (ii) For  $u > u_c$ ,  $|S^*(u)| = o_P(n)$ .
- (iii)  $\phi^*(u) - \phi_c \sim C * \sqrt{u_c - u}$  as  $u \rightarrow u_c^-$ .

Methodology (multiple angles):

- (A) Random sampling of ER graphs at several sizes  $n$  in  $\{400, 800, 1600, 3200\}$  with multiple independent trials; compute  $\phi_n(u)$  for a grid of  $u$ .
- (B) PHASE TRANSITION CURVE: plot  $\phi_n(u)$  vs  $u$ ; observe discontinuity.
- (C) COUNTEREXAMPLE SEARCH for monotonicity:  $\phi(u)$  must be non-increasing.
- (D) FINITE-SIZE SCALING: curves for larger  $n$  should approach a limit and sharpen near  $u_c$  (first-order-like jump).
- (E) CRITICAL POINT ESTIMATION: identify  $u_c$  from the drop of  $\phi_n(u)$ .
- (F) SQUARE-ROOT FIT: joint non-linear regression  $\phi = \phi_c + C(u_c - u)^\alpha$ .

The conjecture predicts  $\alpha = 1/2$ .

- (G) LOG-LOG SCALING: slope of  $\log(\phi - \phi_c)$  vs  $\log(u_c - u) \sim 0.5$ .
- (H) SUPER-CRITICAL VANISHING:  $|S^*(u)|$  grows sublinearly for  $u > u_c$ .
- (I) ROBUSTNESS across two  $(c, \beta)$  parameter choices.
- (J) EDGE CASES:  $u=1$  (all non-negative Bonacich survive),  $u$  large (empty core), small graphs.

Bonacich vectors computed via Neumann iteration  $x \leftarrow 1 + \beta A_S x$  (convergent when  $\beta * \rho(A_S) < 1$ ) with direct-solve fallback.

""")

```
t_start = time.time()
```

```
RNG = np.random.default_rng(20260420)
```

```
# ----- Core algorithms -----
```

```
def make_er(n, c, rng):
```

```
    p = c / n
```

```
    iu, ju = np.triu_indices(n, k=1)
```

```
    m = rng.random(len(iu)) < p
```

```
    r = np.concatenate([iu[m], ju[m]])
```

```
    co = np.concatenate([ju[m], iu[m]])
```

```
    return sp.csr_matrix((np.ones(len(r)), (r, co)), shape=(n, n))
```

```
def bonacich_vec(A_S, beta, tol=1e-9, max_iter=300):
```

```
    n = A_S.shape[0]
```

```
    if n == 0:
```

```
        return np.zeros(0)
```

```
    x = np.ones(n)
```

```
    diverged = False
```

```
    for _ in range(max_iter):
```

```
        xn = 1.0 + beta * (A_S @ x)
```

```
        if (not np.isfinite(xn).all()) or np.max(np.abs(xn)) > 1e12:
```

```
            diverged = True
```

```
            break
```

```
        if np.max(np.abs(xn - x)) < tol:
```

```
            return xn
```

```
        x = xn
```

```
    if diverged or np.max(np.abs(xn - x)) > 1e-5:
```

```
        try:
```

```
            I = sp.eye(n, format='csc')
```

```
            M = (I - beta * A_S).tocsc()
```

```

        return spsolve(M, np.ones(n))
    except Exception:
        return x
return x

def bonacich_core(A, beta, u, max_outer=None):
    n = A.shape[0]
    if max_outer is None:
        max_outer = max(200, n)
    active = np.ones(n, dtype=bool)
    prev_size = n + 1
    for _ in range(max_outer):
        idx = np.flatnonzero(active)
        k = len(idx)
        if k == 0 or k == prev_size:
            return active
        A_S = A[idx][:, idx]
        b = bonacich_vec(A_S, beta)
        keep = b >= u - 1e-12
        if keep.all():
            return active
        new_active = np.zeros(n, dtype=bool)
        new_active[idx[keep]] = True
        prev_size = k
        active = new_active
    return active

# ----- Part 1: main experiment at (c, beta) = (3.0, 0.2) -----
c, beta = 3.0, 0.2
ns = [400, 800, 1600, 3200]
trials_per_n = [4, 3, 2, 2]
us = np.linspace(1.0, 4.5, 30)

print(f"\nMAIN: c={c}, beta={beta}, ns={ns}, trials={trials_per_n}, |u-grid|={len(us)}")
phi_data = {n: np.zeros((t, len(us))) for n, t in zip(ns, trials_per_n)}

for n, T in zip(ns, trials_per_n):
    for t in range(T):
        A = make_er(n, c, RNG)
        for ui, u in enumerate(us):
            core = bonacich_core(A, beta, u)
            phi_data[n][t, ui] = core.sum() / n
        print(f"n={n:5d} trial={t+1}/{T} t={time.time()-t_start:5.1f}s")

phi_mean = {n: phi_data[n].mean(axis=0) for n in ns}
phi_std = {n: phi_data[n].std(axis=0) for n in ns}

# ----- Part 2: Monotonicity (counterexample search) -----
mono_viol = 0
for n in ns:
    for t in range(phi_data[n].shape[0]):
        diffs = np.diff(phi_data[n][t])
        mono_viol += int((diffs > 1e-9).sum())

```

```

print(f"\nMonotonicity violations across all samples/trials: {mono_viol}")

# ----- Part 3: Identify u_c and phi_c from the largest n curve -----
pm_big = phi_mean[ns[-1]]
# Rough estimate: largest u with phi still > threshold
threshold = 0.02
above = pm_big > threshold
if above.any():
    idx_crit = int(np.max(np.where(above)[0]))
    u_c_rough = us[idx_crit]
    phi_c_rough = pm_big[idx_crit]
else:
    u_c_rough, phi_c_rough = us[0], 0.0
print(f"Rough (u_c, phi_c) from n={ns[-1]}: ({u_c_rough:.3f}, {phi_c_rough:.3f})
# ... [truncated]

```

## 5 Conclusion

The conjecture remains formally open. Numerical experiments were **inconclusive** — neither strong support nor clear counterexamples were found. Further investigation (both formal and empirical) is warranted.