## A few random open problems, some of them for random graphs

#### Paweł Prałat

#### Department of Mathematics, Ryerson University, Toronto, ON

#### Workshop "Probability and Graphs" — January 2014

イロメ イ押メ イヨメ イヨメー

<span id="page-0-0"></span>B

#### **Outline**



2 [Total acquisition in random graphs](#page-20-0)

3 Cops [and Robbers on Boolean lattice](#page-38-0)



4 0 8  $\leftarrow$  $\mathbf{p}$ 

医电子 化重子

B

#### **Outline**



[Total acquisition in random graphs](#page-20-0)

**Cops [and Robbers on Boolean lattice](#page-38-0)** 



4 0 8

 $\left\{ \bigoplus_k k \right\} \in \mathbb{R}$  is a defined of

<span id="page-2-0"></span>ă,

#### Place one *agent* on each vertex.

- Every pair of agents sharing an edge is declared to be *acquainted*, and remains so throughout the process.
- In each round, we chose some matching *M* (any matching, perhaps it is a single edge). For each edge of *M*, we swap the agents occupying its endpoints.
- The *acquaintance time*, *AC*(*G*), is the minimum number of rounds required for all agents to become acquainted with one another.

K ロ ⊁ K 何 ≯ K ヨ ⊁ K ヨ ⊁

- Place one *agent* on each vertex.
- Every pair of agents sharing an edge is declared to be *acquainted*, and remains so throughout the process.
- In each round, we chose some matching *M* (any matching, perhaps it is a single edge). For each edge of *M*, we swap the agents occupying its endpoints.
- The *acquaintance time*, *AC*(*G*), is the minimum number of rounds required for all agents to become acquainted with one another.

K ロ ⊁ K 何 ≯ K ヨ ⊁ K ヨ ⊁

- Place one *agent* on each vertex.
- Every pair of agents sharing an edge is declared to be *acquainted*, and remains so throughout the process.
- In each round, we chose some matching *M* (any matching, perhaps it is a single edge). For each edge of *M*, we swap the agents occupying its endpoints.
- The *acquaintance time*, *AC*(*G*), is the minimum number of rounds required for all agents to become acquainted with one another.

イロメ イ何 メイヨメ イヨメ

- **Place one** *agent* on each vertex.
- Every pair of agents sharing an edge is declared to be *acquainted*, and remains so throughout the process.
- In each round, we chose some matching *M* (any matching, perhaps it is a single edge). For each edge of *M*, we swap the agents occupying its endpoints.
- The *acquaintance time*, *AC*(*G*), is the minimum number of rounds required for all agents to become acquainted with one another.

イロメ イ何 メイヨメ イヨメ

B

- Place one *agent* on each vertex.
- Every pair of agents sharing an edge is declared to be *acquainted*, and remains so throughout the process.
- In each round, we chose some matching *M* (any matching, perhaps it is a single edge). For each edge of *M*, we swap the agents occupying its endpoints.
- The *acquaintance time*, *AC*(*G*), is the minimum number of rounds required for all agents to become acquainted with one another.

Clearly,

$$
AC(G) \ge \frac{{|V(G)| \choose 2}}{|E(G)|} - 1
$$

4 ロ ) 4 何 ) 4 ミ ) 4 ミ )

### How large can this be for *n*-vertex connected graph?

Theorem (Benjamini, Shinkar, Tsur, 2014+)

$$
AC(G) = O\left(\frac{n^2}{\log n/\log\log n}\right)
$$

$$
AC(G) = O\left(\frac{n^2}{\log n}\right)
$$

$$
AC(G) = O\left(n^{3/2}\right)
$$

(This result is tight.)

イロト イ押 トイヨ トイヨ トーヨー

## How large can this be for *n*-vertex connected graph?

Theorem (Benjamini, Shinkar, Tsur, 2014+)

$$
AC(G) = O\left(\frac{n^2}{\log n/\log\log n}\right)
$$

Theorem (Kinnersley Mitsche, Pralat, 2013)

$$
AC(G) = O\left(\frac{n^2}{\log n}\right)
$$

$$
AC(G) = O\left(n^{3/2}\right)
$$

(This result is tight.)

**K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** 

÷.  $QQ$ 

## How large can this be for *n*-vertex connected graph?

Theorem (Benjamini, Shinkar, Tsur, 2014+)

$$
AC(G) = O\left(\frac{n^2}{\log n/\log\log n}\right)
$$

Theorem (Kinnersley Mitsche, Pralat, 2013)

$$
AC(G) = O\left(\frac{n^2}{\log n}\right)
$$

Theorem (Angel, Shinkar, 2014+)

$$
AC(G) = O\left(n^{3/2}\right)
$$

(This result is tight.)

イロト イ押 トイヨ トイヨ トー

÷.  $QQ$ 

**K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** 

÷.  $QQ$ 

## Random graphs *G*(*n*, *p*)

Conjecture (Benjamini, Shinkar, Tsur, 2014+)

For  $p = p(n) > (\log n + \omega)/n$  we have a.a.s.

$$
AC(G) = O\left(\frac{polylog(n)}{p}\right)
$$

They conjectured this despite the fact that no non-trivial upper bound was known.

Trivial upper bound:  $O(n)$ , provided that  $\omega - \log \log n \to \infty$ (since  $AC(G) = O(n)$  for any graph with Hamiltonian path).

Trivial lower bound: Ω(*n* <sup>2</sup>/*m*) = Ω(1/*p*).

 $2Q$ 

## Random graphs *G*(*n*, *p*)

Theorem (Kinnersley Mitsche, Pralat, 2013)

*For*  $p = p(n) \geq (\log n + \log \log n + \omega)/n$  *we have a.a.s.* 

$$
AC(G) = O\left(\frac{\log n}{p}\right)
$$

Is it tight?

*AC*(*G*): each agent has a helicopter and can, on each round, move to any vertex she wants. Clearly,

$$
AC(G) \geq \overline{AC}(G)
$$

*AC*(*G*) also represents the minimum number of copies of a graph *G* needed to cover all edges of a complete graph of the same order. イロト イ団ト イヨト イヨト B

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Theorem (Kinnersley Mitsche, Pralat, 2013)

*For*  $p = p(n) \geq (\log n + \log \log n + \omega)/n$  *we have a.a.s.* 

$$
AC(G) = O\left(\frac{\log n}{p}\right)
$$

Is it tight?

*AC*(*G*): each agent has a helicopter and can, on each round, move to any vertex she wants. Clearly,

$$
AC(G) \geq \overline{AC}(G)
$$

*AC*(*G*) also represents the minimum number of copies of a graph *G* needed to cover all edges of a complete graph of the same order. **K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** ÷.

K ロ ▶ K @ ▶ K 할 ▶ K 할 ▶ ① 할 → ⊙ Q @

## Random graphs *G*(*n*, *p*)

Theorem (Kinnersley Mitsche, Pralat, 2013)

*For*  $p = p(n)$  *>* (log  $n + \log \log n + \omega$ )/*n* we have a.a.s.

$$
AC(G) = O\left(\frac{\log n}{p}\right)
$$

Theorem (Kinnersley Mitsche, Pralat, 2013)

*Let*  $\varepsilon > 0$ *. For*  $p = p(n) \ge n^{-1/2 + \varepsilon}$  and  $p \le 1 - \varepsilon$  we have a.a.s.

$$
\overline{AC}(G)=\Theta(AC(G))=\Theta\left(\frac{\log n}{\rho}\right)
$$

The [acquaintance time](#page-2-0) [Total acquisition in random graphs](#page-20-0) [Cops and Robbers on Boolean lattice](#page-38-0) [Lazy Cops and Robbers](#page-46-0)

イロト イ押 トイヨ トイヨ トーヨー

 $2Q$ 

## Random graphs *G*(*n*, *p*)

Theorem (Kinnersley Mitsche, Pralat, 2013)

*For*  $p = p(n) \geq (log n + log log n + \omega)/n$  *we have a.a.s.* 

$$
AC(G) = O\left(\frac{\log n}{\rho}\right)
$$

#### Theorem (Kinnersley Mitsche, Pralat, 2013)

*Let*  $\varepsilon > 0$ *. For*  $p = p(n) \ge n^{-1/2 + \varepsilon}$  and  $p \le 1 - \varepsilon$  we have a.a.s.

$$
\overline{AC}(G) = \Theta(AC(G)) = \Theta\left(\frac{\log n}{p}\right)
$$

The behaviours of *AC*(*G*) and *AC*(*G*) for sparse random graphs remain undetermined!

K ロ ▶ K @ ▶ K 할 ▶ K 할 ▶ ① 할 → ⊙ Q @

## Random graphs *G*(*n*, *p*)

Theorem (Kinnersley Mitsche, Pralat, 2013)

*For*  $p = p(n) \geq (log n + log log n + \omega)/n$  *we have a.a.s.* 

$$
AC(G) = O\left(\frac{\log n}{\rho}\right)
$$

#### Theorem (Kinnersley Mitsche, Pralat, 2013)

*Let*  $\varepsilon > 0$ *. For*  $p = p(n) \ge n^{-1/2 + \varepsilon}$  and  $p \le 1 - \varepsilon$  we have a.a.s.

$$
\overline{AC}(G) = \Theta(AC(G)) = \Theta\left(\frac{\log n}{p}\right)
$$

#### Check what is going on below the threshold for Hamiltonian path and when  $p \rightarrow 1$ ?

#### Random geometric graphs *G*(*n*, *r*)

#### Theorem (Muller, Pralat, 2014+)

*If*  $\pi$ *nr*<sup>2</sup> – log *n*  $\rightarrow \infty$ , then a.a.s.

$$
AC(G)=\Theta(r^{-2}).
$$

(Recall that if  $\pi nr^2 - \log n \to -\infty$ , then *G* is a.a.s. disconnected and so the acquaintance time is not defined.)

イロト イ押 トイヨ トイヨ トー

B

## Percolated random geometric graphs *G*(*n*, *r*, *p*)

#### Theorem (Muller, Pralat, 2014+)

Let  $\varepsilon > 0$ . If  $pnr^2 \geq n^{1/2+\varepsilon}$ , then a.a.s.

$$
AC(G) = \Theta(r^{-2}p^{-1}\log n).
$$

#### (In fact, an upper bound works whenever  $pnr^2 > K$  log *n* for some large constant *K*.)

イロン イ何ン イヨン イヨン 一ヨー

 $200$ 

### Percolated random geometric graphs *G*(*n*, *r*, *p*)

Theorem (Muller, Pralat, 2014+)

Let  $\varepsilon > 0$ . If  $pnr^2 \geq n^{1/2+\varepsilon}$ , then a.a.s.

$$
AC(G) = \Theta(r^{-2}p^{-1}\log n).
$$

(In fact, an upper bound works whenever  $pnr^2 > K$  log *n* for some large constant *K*.)

The behaviour for sparser graphs remains undetermined.

**≮ロト ⊀伊 ▶ ⊀ ヨ ▶ ⊀ ヨ ▶** 

÷.  $QQ$ 

### **Outline**



#### 2 [Total acquisition in random graphs](#page-20-0)

#### **Cops [and Robbers on Boolean lattice](#page-38-0)**



4 ロ ) 4 何 ) 4 ミ ) 4 ミ )

 $2990$ 

<span id="page-20-0"></span>ă,

#### • Initially, each vertex has weight 1.

- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

4 何 8

→ 唐 > → 唐 >

 $\Omega$ 

- Initially, each vertex has weight 1.
- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

K 御 ▶ K ヨ ▶ K ヨ ▶

- Initially, each vertex has weight 1.
- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

- 4 周 8 3 4 周 8

- Initially, each vertex has weight 1.
- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

Clearly,  $a_t(G) < \gamma(G)$ .

K 何 ▶ K ヨ ▶ K ヨ ▶

- Initially, each vertex has weight 1.
- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

Clearly,  $a_t(G) < \gamma(G)$ .

Also,  $a_t(G) > a_t(H)$  if *G* is a subgraph of *H* (on the same vertex set).

イロメ 不優 トイヨメ イヨメー

ă,  $QQ$ 

- $\bullet$  Initially, each vertex has weight 1.
- In each step, we are allowed to move weight from a vertex *u* to a neighbouring vertex *v*, provided that before the move the weight on *v* is at least the weight on *u*.
- The *total acquisition number at*(*G*) is the minimum possible size of the final set of vertices with positive weight.

Clearly,  $a_t(G) \leq \gamma(G)$ .

Also,  $a_t(G) > a_t(H)$  if *G* is a subgraph of *H* (on the same vertex set).

So  $a_t(G(n, p_1)) \leq a_t(G(n, p_2))$ , provided  $p_1 > p_2$ .

イロト イ押 トイヨ トイヨ トー

÷.  $QQ$  The [acquaintance time](#page-2-0) [Total acquisition in random graphs](#page-20-0) [Cops and Robbers on Boolean lattice](#page-38-0) [Lazy Cops and Robbers](#page-46-0)

### Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n-1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

イロト イ押 トイヨ トイヨ トー

÷.

イロト イ押 トイヨ トイヨト

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

**•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So *p* = 1/2 works!

イロメ 不優 トイヨメ イヨメー

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So *p* = 1/2 works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).

### Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min{ $a_t(G), a_t(\bar{G})$ } = 1 (LeSaulnier, Prince, Wenger, West, Worah, 2013). So  $p = 1/2$  works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = C \log^2 n / \log \log n$  (Bal, Bennett, Dudek, Pralat,  $\approx 40$ days ago).

**◆ロ→ ◆伊→ ◆ミ→ → ミ→ → ミ** 

**≮ロト ⊀伊 ▶ ⊀ ヨ ▶ ⊀ ヨ ▶** 

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So *p* = 1/2 works!
- *d* = *n*<sup>ε</sup> (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $\bullet$  *d* = *C* log *n* (Bal, Bennett, Dudek, Pralat, ≈ 25 days ago).

### Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So  $p = 1/2$  works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\log 2}$ <u>1+∈</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

**K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** 

イロト イ押 トイヨ トイヨ トーヨー

 $2990$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min{ $a_t(G), a_t(\bar{G})$ } = 1 (LeSaulnier, Prince, Wenger, West, Worah, 2013). So *p* = 1/2 works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\log 2}$ <u>li+a</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

Note that  $\frac{1}{\log 2} \approx 1.44$  so it is above the connectivity threshold.

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(\overline{G})\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So *p* = 1/2 works!
- *d* = *n*<sup>ε</sup> (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\log 2}$ <u>l +є</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

Sharp! For  $d = \frac{1-\varepsilon}{\log 2}$ log 2 log *n*, a.a.s. *at*(*G*(*n*, *p*)) > *n* ε+*o*(1) (Bal, Bennett, Dudek, Pralat,  $\approx$  25 days ago).

**◆ロ→ ◆伊→ ◆ミ→ → ミ→ → ミ** 

イロト イ押 トイヨ トイヨ トーヨー

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So  $p = 1/2$  works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\ln 2}$ <u>1+∈</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

Sharp! For  $d = \frac{1-\varepsilon}{\log 2}$ log 2 log *n*, a.a.s. *at*(*G*(*n*, *p*)) > *n* ε+*o*(1) (Bal, Bennett, Dudek, Pralat,  $\approx$  25 days ago).

 $Investigate  $f(c) = log_a a_t(G(n, c \log n/n))$ .$ 

**◆ロ→ ◆伊→ ◆ミ→ → ミ→ → ミ** 

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So  $p = 1/2$  works!
- $d = n^{\varepsilon}$  (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\ln 2}$ <u>1+∈</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

Sharp! For  $d = \frac{1-\varepsilon}{\log 2}$ log 2 log *n*, a.a.s. *at*(*G*(*n*, *p*)) > *n* ε+*o*(1) (Bal, Bennett, Dudek, Pralat,  $\approx$  25 days ago).

Investigate what is going on when  $d = \frac{1+o(1)}{\log 2}$ log 2 log *n*.

イロト イ押 トイヨ トイヨ トーヨー

 $QQ$ 

## Random graphs *G*(*n*, *p*)

Question (LeSaulnier, Prince, Wenger, West, Worah, 2013): Find the smallest  $d = p(n - 1)$  such that a.a.s.  $a_t(G(n, p)) = 1$ .

- **•** If  $G \neq C_5$ , then min $\{a_t(G), a_t(G)\} = 1$  (LeSaulnier, Prince, Wenger, West, Worah, 2013). So  $p = 1/2$  works!
- *d* = *n*<sup>ε</sup> (Krivelevich, 2010, Embedding spanning trees in random graphs).
- $d = \frac{1+\varepsilon}{\ln 2}$ <u>1+∈</u> log *n* (Bal, Bennett, Dudek, Pralat, ≈ 5 days ago).

Sharp! For  $d = \frac{1-\varepsilon}{\log 2}$ log 2 log *n*, a.a.s. *at*(*G*(*n*, *p*)) > *n* ε+*o*(1) (Bal, Bennett, Dudek, Pralat,  $\approx$  25 days ago).

What about other models, for example, random geometric graphs?

#### **Outline**



[Total acquisition in random graphs](#page-20-0)

3 Cops [and Robbers on Boolean lattice](#page-38-0)



4 ロ ) 4 何 ) 4 ミ ) 4 ミ )

<span id="page-38-0"></span>B



- The robber starts at 'level' 0, vertex  $(0,0,\ldots,0)$ .
- Cops start at 'level' n, vertex  $(n, n, \ldots, n)$ .

イロメ 不優 トイヨメ イヨメー

 $2990$ 

÷.

- The robber starts at 'level' 0, vertex  $(0,0,\ldots,0)$ .
- Cops start at 'level' n, vertex  $(n, n, \ldots, n)$ .
- At each step, the robber goes up and cops go down.

イロメ イ押 メイヨメ イヨメ

÷.

- The robber starts at 'level' 0, vertex  $(0, 0, \ldots, 0)$ .
- Cops start at 'level' n, vertex  $(n, n, \ldots, n)$ .
- At each step, the robber goes up and cops go down.
- *c*(*n*) is the minimum number of cops needed to catch the robber.

イロメ イ何 メイヨメ イヨメ

÷.

- $\bullet$  The robber starts at 'level' 0, vertex  $(0,0,\ldots,0)$ .
- Cops start at 'level' n, vertex  $(n, n, \ldots, n)$ .
- At each step, the robber goes up and cops go down.
- *c*(*n*) is the minimum number of cops needed to catch the robber.

Greedy strategy for the robber gives:

$$
c(n) \ge 2^m = 2^{n/2}, \text{ provided } n = 2m
$$
  

$$
c(n) \ge 2^{-m} {2m + 1 \choose m + 1} = \Theta(2^{n/2}/\sqrt{n}), \text{ provided } n = 2m + 1
$$

イロト イ押 トイヨ トイヨ トー

÷.  $QQ$ 

イロメ イ何 メイヨメ イヨメ

 $QQ$ 

## **Definition**

- $\bullet$  The robber starts at 'level' 0, vertex  $(0,0,\ldots,0)$ .
- Cops start at 'level' n, vertex (*n*, *n*, . . . , *n*).
- At each step, the robber goes up and cops go down.
- *c*(*n*) is the minimum number of cops needed to catch the robber.

Greedy strategy for the robber gives:

$$
c(n) \ge 2^m = 2^{n/2}, \text{ provided } n = 2m
$$
  

$$
c(n) \ge 2^{-m} {2m + 1 \choose m + 1} = \Theta(2^{n/2}/\sqrt{n}), \text{ provided } n = 2m + 1
$$

Is there any upper bound?

Greedy strategy for the robber gives:

$$
c(n) \ge 2^m = 2^{n/2}, \text{ provided } n = 2m
$$
  

$$
c(n) \ge 2^{-m} {2m + 1 \choose m + 1} = \Theta(2^{n/2}/\sqrt{n}), \text{ provided } n = 2m + 1
$$

#### Is there any upper bound?

For the upper bound, cops can flip random bits (independently) to get an upper bound that is larger by a factor of *n* log *n*.

4 0 8

 $\langle \oplus \rangle$  >  $\langle \oplus \rangle$  >  $\langle \oplus \rangle$ 

Greedy strategy for the robber gives:

$$
c(n) \ge 2^m = 2^{n/2}, \text{ provided } n = 2m
$$
  

$$
c(n) \ge 2^{-m} {2m + 1 \choose m + 1} = \Theta(2^{n/2}/\sqrt{n}), \text{ provided } n = 2m + 1
$$

#### Is there any upper bound?

For the upper bound, cops can flip random bits (independently) to get an upper bound that is larger by a factor of *n* log *n*.

#### Where is the right value?

4 0 8

K 何 ▶ K ヨ ▶ K ヨ ▶

#### **Outline**



[Total acquisition in random graphs](#page-20-0)

**Cops [and Robbers on Boolean lattice](#page-38-0)** 



4 0 8

 $\left\{ \bigoplus_k k \right\} \in \mathbb{R}$  is a defined of

 $2990$ 

<span id="page-46-0"></span>ă,

#### Classic game

- *G*(*n*, *p*): dense graphs (Bonato, Pralat, Wang, 2009)
- *G*(*n*, *p*): very dense graphs (Pralat, 2010)
- *G*(*n*, *p*): zig-zag (Luczak, Pralat, 2010)
- *G*(*n*, *p*): sparse graphs (Bollobas, Kun, Leader, 2013)
- *G*(*n*, *p*): Meyniel's conjecture (Pralat, Wormald, 2014+)
- **•** random *d*-regular graphs: Meyniel's conjecture (Pralat, Wormald, 2014++)
- *G*(*n*, *r*): (Beveridge, Dudek, Frieze, Muller, 2012) and (Alon, Pralat, 2014+)
- *G*(*n*, *r*, *p*): (Alon, Pralat, 2014+)

K ロ ▶ K 御 ▶ K ヨ ▶ K ヨ ▶ ...

ミー  $QQ$ 

イロト イ押 トイヨ トイヨト

÷.  $QQ$ 

### Other variants for *G*(*n*, *p*)

- Fast robber (Alon, Mehrabian, 2014+)
- Playing on edges (Dudek, Gordinowicz, Pralat, 2014+)
- Cops have limited fuel or time (Fomin, Golovach, Pralat, 2012)
- Cops can shoot from distance (Bonato, Chiniforooshan, Pralat, 2010)
- $\bullet$  ...
- Lazy Cops (Bal, Bonato, Kinnersley, and Pralat, 2014+)

$$
c_L(Q_n)=O\left(\tfrac{2^n\log n}{n^{3/2}}\right)
$$

(Offner and Ojakian, 2014+) 
$$
c_L(Q_n) = \Omega\left(2^{\sqrt{n}/20}\right)
$$

(Bal, Bonato, Kinnersley, and Pralat, 2014+) For any  $\varepsilon > 0$ ,  $c_L(Q_n) = \Omega\left(\frac{2^n}{n^{7/2}}\right)$  $\frac{2^n}{n^{7/2+\varepsilon}}$ 

Where is the right value?

メロメメ 御きメ ミトメ ヨメー

 $299$ 

÷.

$$
c_L(Q_n)=O\left(\tfrac{2^n\log n}{n^{3/2}}\right)
$$

#### (Offner and Ojakian, 2014+)  $c_{L}(Q_{n})=\Omega\left(2\right)$  $\sqrt{n}/20$

(Bal, Bonato, Kinnersley, and Pralat, 2014+) For any  $\varepsilon > 0$ ,  $c_L(Q_n) = \Omega\left(\frac{2^n}{n^{7/2}}\right)$  $\frac{2^n}{n^{7/2+\varepsilon}}$ 

Where is the right value?

イロメ 不優 トイヨメ イヨメー

B

$$
c_L(Q_n)=O\left(\tfrac{2^n\log n}{n^{3/2}}\right)
$$

(Offner and Ojakian, 2014+) 
$$
c_L(Q_n) = \Omega(2^{\sqrt{n}/20})
$$

(Bal, Bonato, Kinnersley, and Pralat, 2014+) For any  $\varepsilon > 0$ ,  $c_L(Q_n) = \Omega\left(\frac{2^n}{n^{7/2}}\right)$  $\frac{2^n}{n^{7/2+\varepsilon}}$ 

Where is the right value?

K ロ ト K 個 ト K 君 ト K 君 ト .

÷.

$$
c_L(Q_n)=O\left(\tfrac{2^n\log n}{n^{3/2}}\right)
$$

(Offner and Ojakian, 2014+) 
$$
c_L(Q_n) = \Omega(2^{\sqrt{n}/20})
$$

(Bal, Bonato, Kinnersley, and Pralat, 2014+) For any  $\varepsilon > 0$ ,  $c_L(Q_n) = \Omega\left(\frac{2^n}{n^{7/2}}\right)$  $\frac{2^n}{n^{7/2+\varepsilon}}$ 

Where is the right value?

**K ロ ト K 何 ト K ヨ ト K ヨ ト** 

÷.

イロト 不優 トイモト 不思 トー

重。 

# *G*(*n*, *p*)—Big picture



**K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** 

 $\equiv$  .  $299$ 

## *G*(*n*, *p*)—More details



The ratio between upper and lower bounds:

a) 
$$
x = 1 : 1 + o(1)
$$
.  
\nb)  $\frac{1}{2} < x < 1 : \Theta(1)$ .  
\nc)  $\frac{1}{j+1} < x < \frac{1}{j}$  for some  $j \ge 2 : \Theta(\log n)$ .  
\nd)  $x = \frac{1}{j}$  for some  $j \ge 2 : \Theta(\log^3 n)$ .

イロト イ押 トイヨ トイヨ トー

<span id="page-55-0"></span>重。  $299$ 

## *G*(*n*, *p*)—More details



The ratio between upper and lower bounds:

\n- a) 
$$
x = 1 : 1 + o(1)
$$
.
\n- b)  $\frac{1}{2} < x < 1 : \Theta(1)$ .
\n- c)  $\frac{1}{j+1} < x < \frac{1}{j}$  for some  $j \geq 2 : \Theta(\log n)$ .
\n- d)  $x = \frac{1}{j}$  for some  $j \geq 2 : \Theta(\log^3 n)$ .
\n- e) nothing is done for  $x = 0$  (sparse graphs)
\n

**K ロ ▶ K 何 ▶ K ヨ ▶ K ヨ ▶** 

■ 1  $299$ 

## *G*(*n*, *p*)—More details



The ratio between upper and lower bounds:

\n- a) 
$$
x = 1 : 1 + o(1)
$$
.
\n- b)  $\frac{1}{2} < x < 1 : \Theta(1)$ .
\n- c)  $\frac{1}{j+1} < x < \frac{1}{j}$  for some  $j \geq 2 : \Theta(\log n)$ .
\n- d)  $x = \frac{1}{j}$  for some  $j \geq 2 : \Theta(\log^3 n)$ .
\n- e) nothing is done for  $x = 0$  (sparse graphs) improve the ratio?
\n