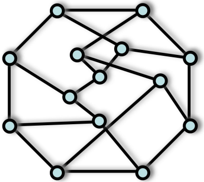
# Random Regular Graphs and Differential Equations

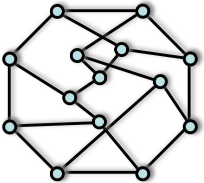
Nick Wormald
University of Waterloo

Minicourse Conference on Random Graph Processes Austin

# Regular graphs - ‘Uniform’ model



# Regular graphs - 'Uniform' model



$\mathcal{G}_{n,d}$ : probability space of $d$-regular graphs on vertex set $\{1, \ldots, n\}$ with uniform distribution:

$$\mathbf{P}(G) = \frac{1}{|\mathcal{G}_{n,d}|} \quad \text{for all} \quad G \in \mathcal{G}_{n,d}.$$

# Some properties of interest

What properties does $G \in \mathcal{G}_{n,d}$ have with respect to

- connectivity, subgraphs?
- Hamilton cycles?
- vertex and edge colourings?
- large independent sets?
- min and max bisections?

# Some properties of interest

What properties does $G \in \mathcal{G}_{n,d}$ have with respect to

- connectivity, subgraphs?
- Hamilton cycles?
- vertex and edge colourings?
- large independent sets?
- min and max bisections?

A property Q holds asymptotically almost surely (a.a.s.) in a random graph model if

$$\mathbf{P}(G \text{ has } Q) \to 1 \text{ as } n \to \infty.$$

# Framework of analysis of random regular graphs

**Theorem [Bender, Canfield '78]**

$$|\mathcal{G}_{n,d}| \sim \frac{(dn)!\,e^{(1-d^2)/4}}{(dn/2)!\,2^{dn/2}\,d!^n}$$

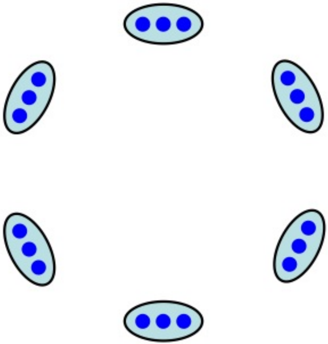
# Framework of analysis of random regular graphs

**Theorem [Bender, Canfield '78]**

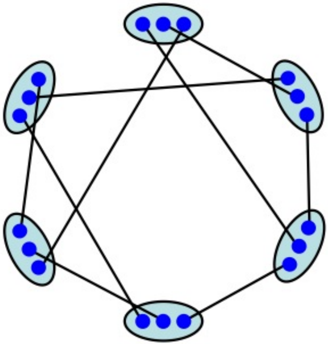
$$|\mathcal{G}_{n,d}| \sim \frac{(dn)!e^{(1-d^2)/4}}{(dn/2)!2^{dn/2}d!^n}$$


Configuration model presented by Béla Bollobás ('79) is convenient for directly showing a.a.s. results.

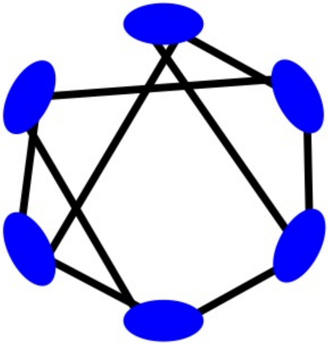
# How to calculate with random regular graphs



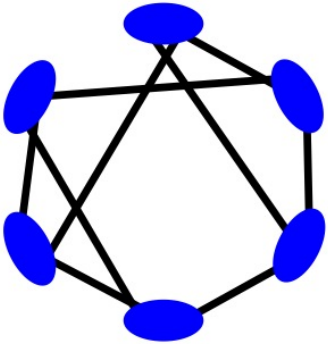
# How to calculate with random regular graphs



# How to calculate with random regular graphs



# How to calculate with random regular graphs



... a random 3-regular graph.

By showing the probability of the graph being simple is asymptotic to $e^{(1-d^2)/4}$, we get the Bender-Canfield formula

$$|\mathcal{G}_{n,d}| \sim \frac{(dn)!\,e^{(1-d^2)/4}}{(dn/2)!\,2^{dn/2}\,d!^n}$$

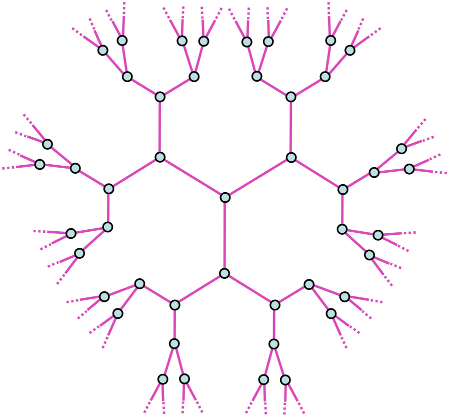
since each simple graph corresponds to $d!^n$ pairings.

## Basic facts on cycles

**Theorem [W '80; Bollobás '80]**

Let $G \in \mathcal{G}_{n,d}$. Let $X_i$ denote the number of cycles of length $i$. Then $X_3, X_4, \ldots$ are asymptotically independent Poisson random variables with means $\mathbf{E}(X_i) \to \frac{(d-1)^i}{2i}$.

Closeup of a large random 3-regular graph:



# Independent sets

Let $\alpha(G)$ denote the independence number of $G$, i.e.

$$\alpha(G) = \text{max cardinality of an independent set of vertices in } G$$

# Independent sets

Let $\alpha(G)$ denote the independence number of $G$, i.e.

$\alpha(G) =$ max cardinality of an independent set of vertices in $G$

**Theorem [Frieze and Łuczak, '92]**

Fix $d \geq 3$. Then $G \in \mathcal{G}_{n,d}$ a.a.s. satisfies

$$\alpha(G) = \frac{2 \log d}{d}\, n\, (1 + O(\xi))$$

where $\xi \to 0$ as $d \to \infty$.

# Independent sets

Let $\alpha(G)$ denote the independence number of $G$, i.e.

$\alpha(G) =$ max cardinality of an independent set of vertices in $G$

**Theorem [Frieze and Łuczak, '92]**

Fix $d \geq 3$. Then $G \in \mathcal{G}_{n,d}$ a.a.s. satisfies

$$\alpha(G) = \frac{2 \log d}{d} \, n \, (1 + O(\xi))$$

where $\xi \to 0$ as $d \to \infty$.

But this says nothing about $d = 3$ say.

# Finding large independent sets

Greedy algorithms find large independent sets in random $d$-regular graphs.

# Finding large independent sets

Greedy algorithms find large independent sets in random $d$-regular graphs.

**Theorem [W, '95]**

Fix $d \geq 3$ and $\epsilon > 0$. Then $G \in \mathcal{G}_{n,d}$ a.a.s. satisfies

$$\alpha(G) \geq (\beta_1(d) - \epsilon)n$$

where $\beta_1(d) = \frac{1}{2}\left(1 - (d-1)^{-2/(d-2)}\right)$.

Method: Build an independent set by randomly adding a vertex, deleting all its neighbours from the graph, and repeating.

# Finding even larger independent sets

The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.

# Finding even larger independent sets

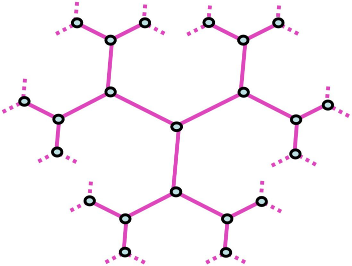
The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.



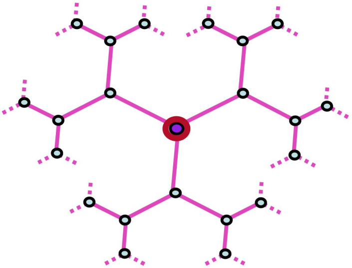
# Finding even larger independent sets

The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:



$G$ is empty.

# Finding even larger independent sets

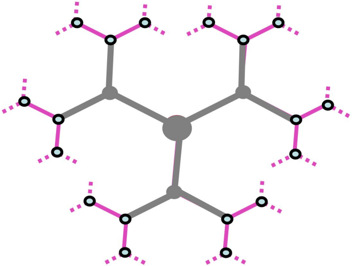
The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.



# Finding even larger independent sets

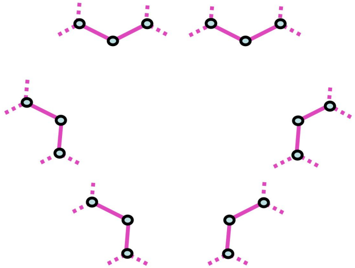
The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.



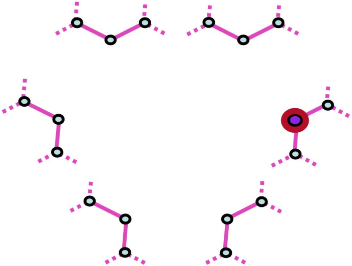
# Finding even larger independent sets

The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:



$G$ is empty.

# Finding even larger independent sets

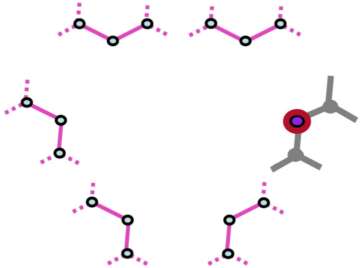
The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.



# Finding even larger independent sets

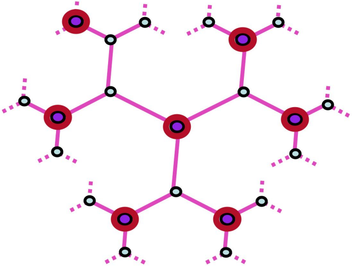
The degree-greedy algorithm for finding an independent set in a graph $G$:

Repeat:

Choose a random vertex $v$ of degree $\delta(G)$.
Add $v$ to the independent set and delete $v$ and its neighbours from $G$.

Until:

$G$ is empty.



# Analysis

We actually apply the degree-greedy algorithm to the pairing model. The remaining pairing remains random (subject to its degree sequence).

Let $Y_i = Y_i(t)$ denote the number of vertices of degree $i$ (i.e. degree counts) in the graph $G_t$ after $t$ steps.

Deleting a vetex of degree $k$ means that the endpoints of $k$ pairs have to be chosen.

The probability that the other end of a pair is in a vertex of degree $j$ is

$$\frac{\text{no. of points in cells of size } j}{\text{total number of points}} = \frac{jY_j + O(1)}{m}$$

where the total number of points is $m = \sum_i iY_i$.

# Expected changes in the variables

Conditioning on the degree counts $\mathbf{Y} = (Y_0, \ldots, Y_d)$,

$$\mathbf{E}\Big(Y_i(t+1) - Y_i(t) \mid \mathbf{Y} \text{ s.t. } \delta(G_t) = r\Big) = f_{i,r}(\mathbf{Y}/n) + o(1).$$

# Expected changes in the variables

Conditioning on the degree counts $\mathbf{Y} = (Y_0, \ldots, Y_d)$,

$$\mathbf{E}\Big(Y_i(t+1) - Y_i(t) \mid \mathbf{Y} \text{ s.t. } \delta(G_t) = r\Big) = f_{i,r}(\mathbf{Y}/n) + o(1).$$

Let $\mathrm{Op}_r$ denote the operation performed when the minimum degree is $r$.

For degree-greedy independent sets algorithm, $\mathrm{Op}_r$ is:

choose a random vertex $v$ of degree $r$
delete $v$ and its neighbours

# Expected changes lead to a d.e.

By studying branching processes we can estimate the proportion $\rho_r$ of steps for which $\mathrm{Op}_r$ is performed (i.e. $\delta(G_t) = r$) in any short segment — depending on $\mathbf{Y}/n$.

This suggests the differential equation

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y})$$

where

$$\mathbf{y} \approx \mathbf{Y}/n, \qquad x = t/n.$$

# Expected changes lead to a d.e.

By studying branching processes we can estimate the proportion $\rho_r$ of steps for which $\mathrm{Op}_r$ is performed (i.e. $\delta(G_t) = r$) in any short segment — depending on $\mathbf{Y}/n$.

This suggests the differential equation

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y})$$

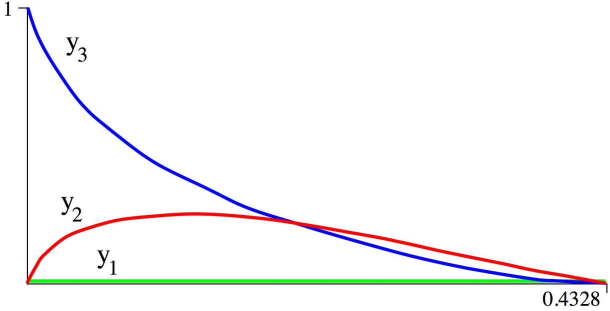
where

$$\mathbf{y} \approx \mathbf{Y}/n, \qquad x = t/n.$$

Comment: $\rho_r(\mathbf{y})$ can be discontinuous, causing phases between non-smooth points, and yielding a system of right-hand derivatives only.

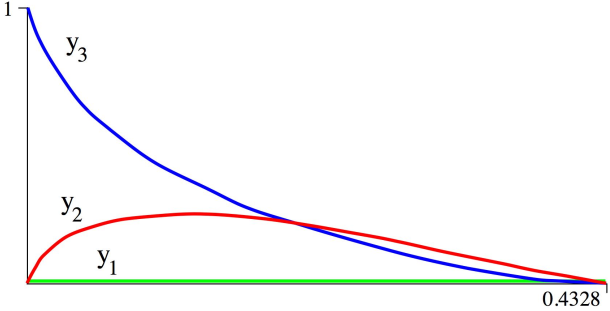
# degree-greedy on 3-regular graphs: solution

Solution of the differential equations:



# degree-greedy on 3-regular graphs: solution

Solution of the differential equations:




We can use a general theorem to show that the process variables a.a.s. track close to the solutions of the differential equations:

# Differential equation method

Key ingredients:

- Boundedness hypothesis:

$$||\mathbf{Y}(t+1) - \mathbf{Y}(t)|| \leq C_0$$

- Trend and Lipschitz hypotheses:

$$||\mathbf{E}(\mathbf{Y}(t+1) - \mathbf{Y}(t) \mid H_t) - f(t/n, \mathbf{Y}(t)/n)| = o(1)$$

  where $H_t$ is the history of the process at time $t$, and $f$ is a Lipschitz function.

Conclusion: the differential equation $\dfrac{d\,\mathbf{y}}{dx} = f(x, \mathbf{y})$ has a unique solution with appropriate initial condition, and a.a.s.

$$\mathbf{Y}(t) = n\,\mathbf{y}(t/n) + o(n)$$

uniformly for $0 \leq t \leq Cn$.

# Lower bounds on independent set ratio

$\beta_0$: earlier known (from Shearer)
$\beta_1$: simple greedy
$\beta_2$: degree greedy

| $d$ | $\beta_0(d)$ | $\beta_1(d)$ | $\beta_2(d)$ |
|---|---|---|---|
| 3 | 0.4139 | 0.3750 | 0.4328 |
| 4 | 0.3510 | 0.3333 | 0.3901 |
| 5 | 0.3085 | 0.3016 | 0.3566 |
| 10 | 0.2032 | 0.2113 | 0.2573 |
| 20 | 0.1297 | 0.1395 | 0.1738 |
| 50 | 0.0682 | 0.0748 | 0.0951 |
| 100 | 0.0406 | 0.0447 | 0.0572 |
| $\to \infty$ | $\to \log d/d$ | $\to \log d/d$ | ? |

# More general issues

Bayati, Gamarnik and Tetali ['13] showed existence of a limiting value for the proportion of vertices in a max independent set ($d$-regular in general, $d$ fixed).

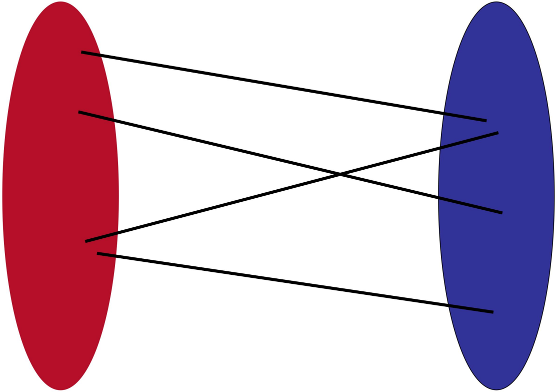
Ding, Sly and Sun recently confirmed the predictions of one-step replica symmetry breaking heuristics to find this limit for $d$ sufficiently large.

# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.

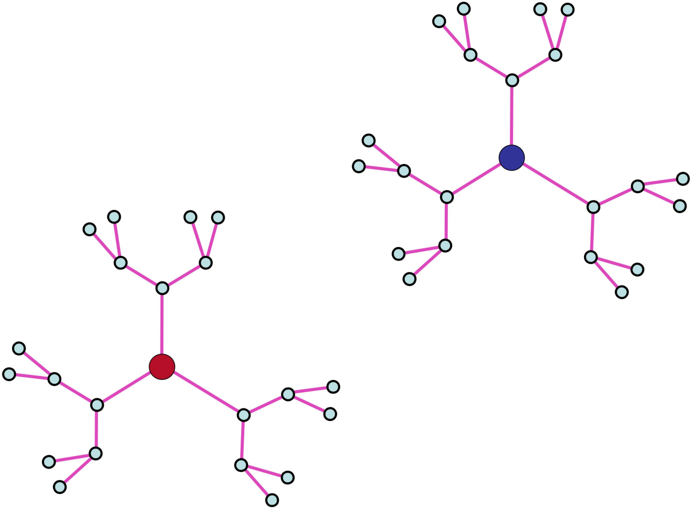
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



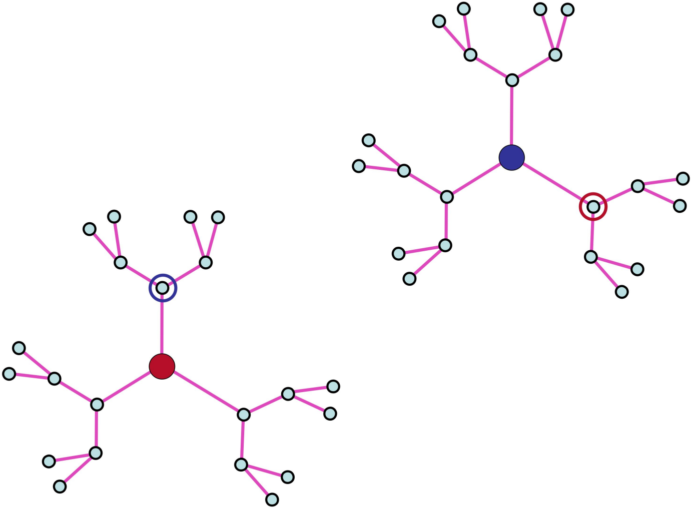
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



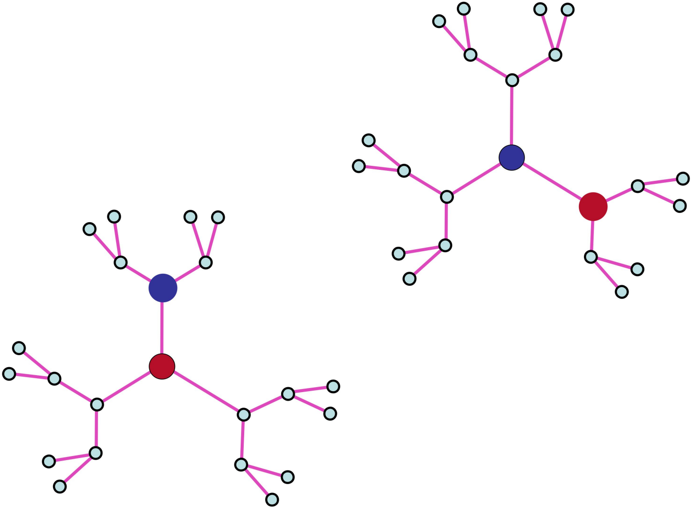
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



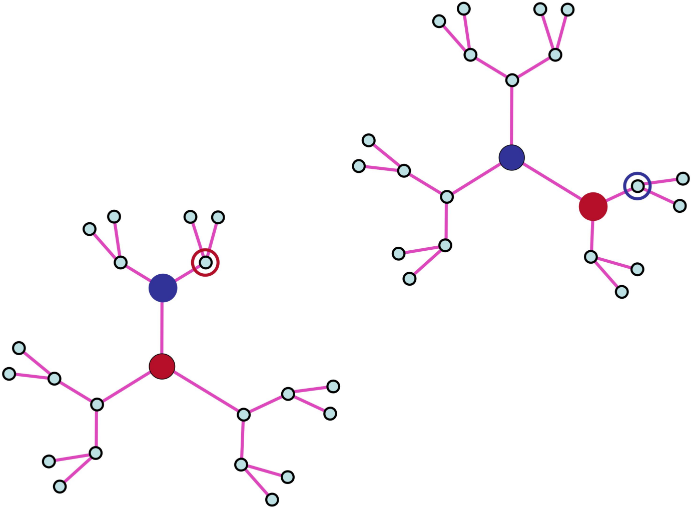
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



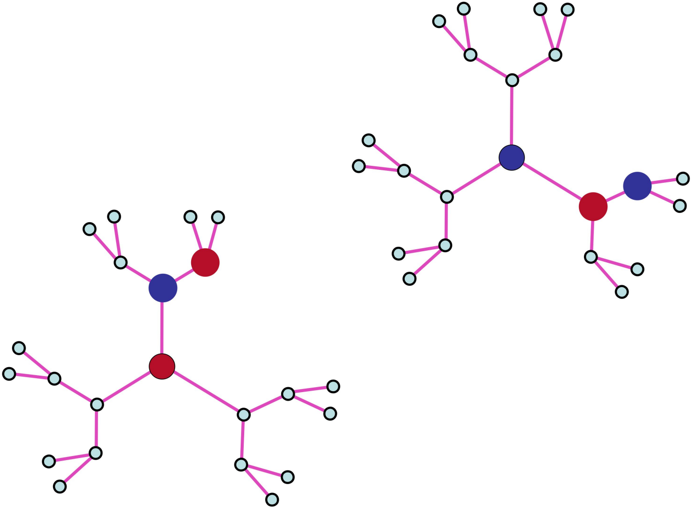
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



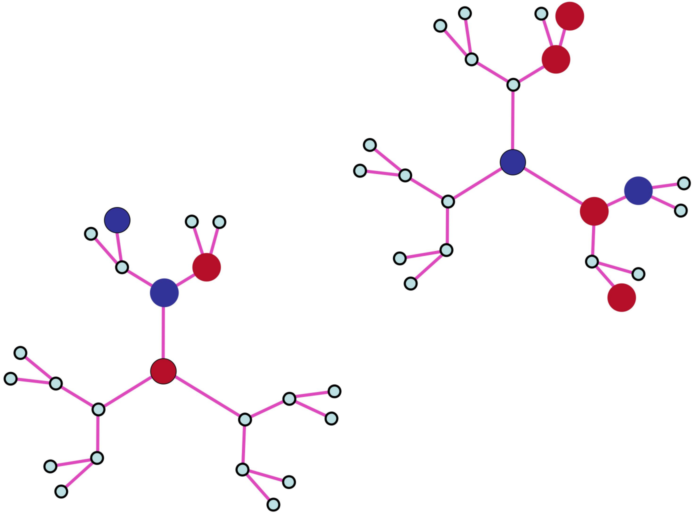
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



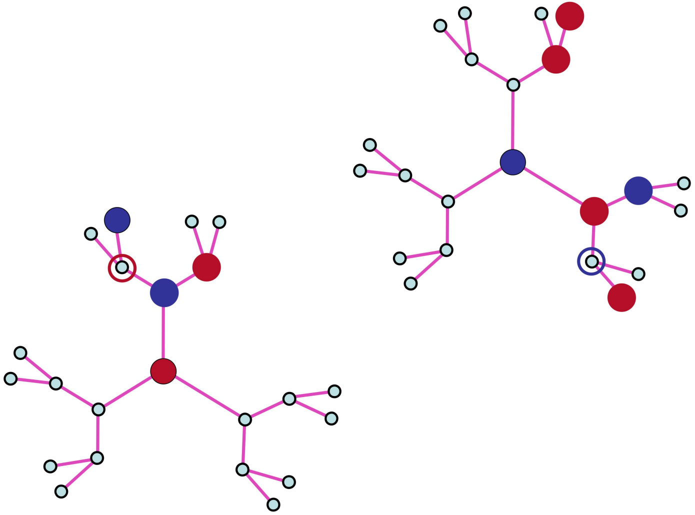
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



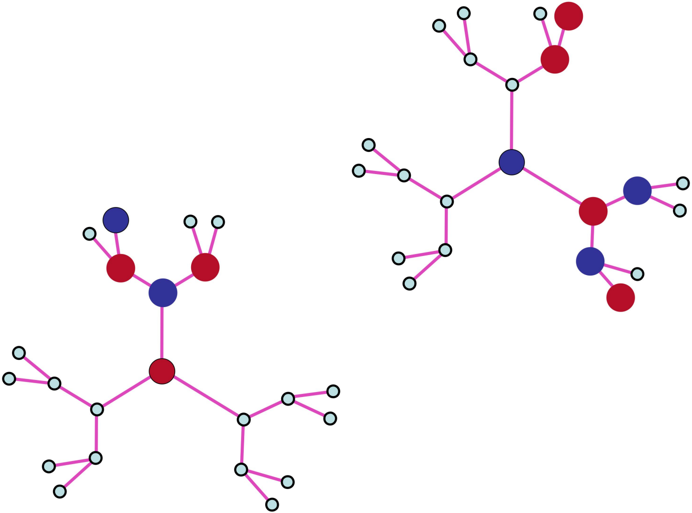
# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



# Max bisection of random regular graphs

Max number of edges crossing a balanced partition of the vertex set.



Lower bounds for max bisection (or max cut) a.a.s. ( Díaz, Do, Serna, W., 2003–2007 ):

3-regular: $1.326n$

4-regular: $5n/3$

5-regular: $1.997n$
etc.

Lower bounds for max bisection (or max cut) a.a.s. ( Díaz, Do, Serna, W., 2003–2007 ):

3-regular: $1.326n$

4-regular: $5n/3$

5-regular: $1.997n$
etc.

Complementary upper bounds for min bisection, i.e.
3-regular$^*$: $0.174n$
4-regular: $n/3$
5-regular: $0.503n$

Lower bounds for max bisection (or max cut) a.a.s. ( Díaz, Do, Serna, W., 2003–2007 ):

3-regular: $1.326n$

4-regular: $5n/3$

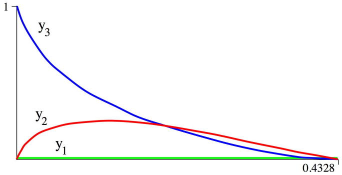
5-regular: $1.997n$
etc.

Complementary upper bounds for min bisection, i.e.
3-regular*: $0.174n$
4-regular: $n/3$
5-regular: $0.503n$

* there are more recent improvements on max cut in the 3-regular case

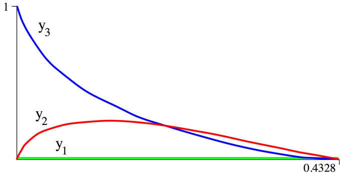
# Deprioritisation for smoother algorithms

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$



# Deprioritisation for smoother algorithms

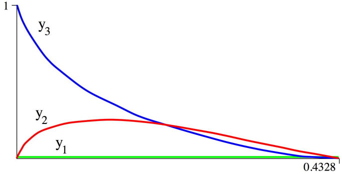
$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$




Same solutions arise if we specify
$\mathbf{P}(\text{op at time } x \text{ is } \mathrm{Op}_r) = \rho_r(\mathbf{y}(x))$.

# Deprioritisation for smoother algorithms

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$

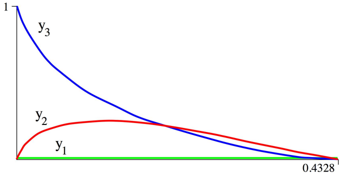



Same solutions arise if we specify
$\mathbf{P}$(op at time $x$ is $\mathrm{Op}_r$) $= \rho_r(\mathbf{y}(x))$.

But the vertices of degree $r$ might become exhausted, prohibiting $\mathrm{Op}_r$.

# Deprioritisation for smoother algorithms

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$



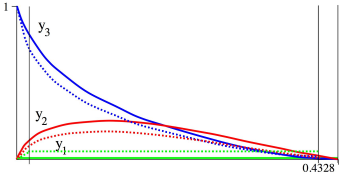

Same solutions arise if we specify
$\mathbf{P}(\text{op at time } x \text{ is } \mathrm{Op}_r) = \rho_r(\mathbf{y}(x))$.

But the vertices of degree $r$ might become exhausted, prohibiting $\mathrm{Op}_r$.

So include an initial period creating vertices of all degrees.

# Deprioritisation for smoother algorithms

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$



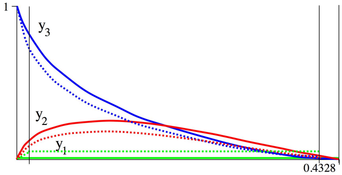

Same solutions arise if we specify
$\mathbf{P}(\text{op at time } x \text{ is } \mathrm{Op}_r) = \rho_r(\mathbf{y}(x))$.

But the vertices of degree $r$ might become exhausted, prohibiting $\mathrm{Op}_r$.

So include an initial period creating vertices of all degrees.

# Deprioritisation for smoother algorithms

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \sum_{r=0}^{d} \rho_r(\mathbf{y}) f_{i,r}(\mathbf{y}) \quad \text{has solutions } y_i(x).$$




Same solutions arise if we specify
$\mathbf{P}(\text{op at time } x \text{ is } \mathrm{Op}_r) = \rho_r(\mathbf{y}(x))$.

But the vertices of degree $r$ might become exhausted, prohibiting $\mathrm{Op}_r$.

So include an initial period creating vertices of all degrees.

The result is a deprioritised algorithm.

# Deprioritised algorithms are convenient

**Theorem [W, '04]**

Provided the functions $f_{i,r}$ governing the prioritised algorithm satisfy certain simple conditions, there is a deprioritised algorithm with behaviour governed by the same differential equation.

# Deprioritised algorithms are convenient

**Theorem [W, '04]**

Provided the functions $f_{i,r}$ governing the prioritised algorithm satisfy certain simple conditions, there is a deprioritised algorithm with behaviour governed by the same differential equation.

This has been convenient to use for a number of algorithms, particularly the ones exhibiting phases.

# Results on other problems

A.a.s. upper bound on minimum independent dominating sets [Duckworth and W., ’06]. (Easiest to analyse using a deprioritised algorithm.)

3-regular: $0.27942n$
4-regular: $0.24399n$
5-regular: $0.21852n$
50-regular: $0.05285n$

## Results on other problems

A.a.s. upper bound on minimum independent dominating sets [Duckworth and W., '06]. (Easiest to analyse using a deprioritised algorithm.)

3-regular: $0.27942n$
4-regular: $0.24399n$
5-regular: $0.21852n$
50-regular: $0.05285n$

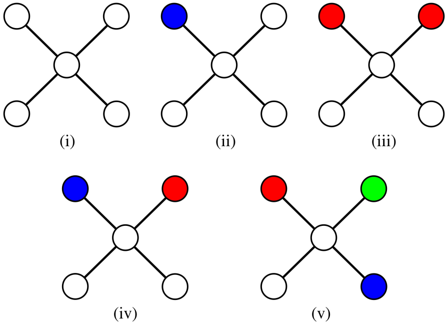
Other results: $k$-independent sets, maximum induced matchings, ...

# Chromatic number of random 4-regular graphs

Greedy colouring algorithm analysed by differential equations.

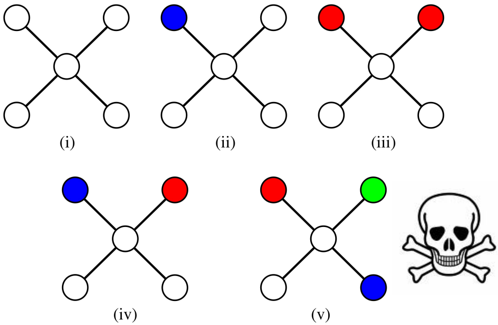
# Chromatic number of random 4-regular graphs

Greedy colouring algorithm analysed by differential equations.



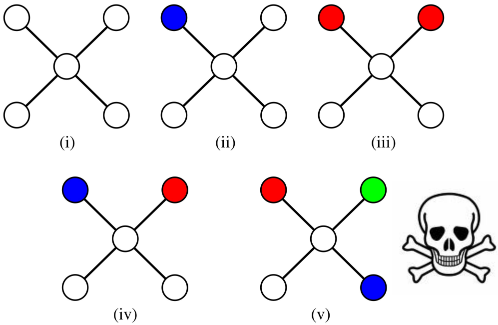
# Chromatic number of random 4-regular graphs

Greedy colouring algorithm analysed by differential equations.



# Chromatic number of random 4-regular graphs

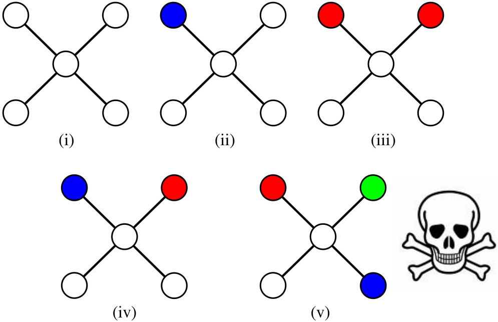
Greedy colouring algorithm analysed by differential equations.




BUT colour the short odd cycles first, THEN be greedy.

# Chromatic number of random 4-regular graphs

Greedy colouring algorithm analysed by differential equations.




BUT colour the short odd cycles first, THEN be greedy.

AND stop with *cn* vertices uncoloured.

**Theorem [Shi and W, '07]**

$\chi(\mathcal{G}_{n,4}) = 3$ a.a.s.

**Theorem [Shi and W, '07]**

$\chi(\mathcal{G}_{n,4}) = 3$ a.a.s.

Similarly,

5-regular: 3 or 4

6-regular: 4

7-regular: 4 or 5

# Just a few unsolved problems

What is the (limiting) size of the

- largest independent set
- max cut
- min bisection

in a random 3-regular graph?

Is there a limiting size (scaled by $n$) in the $d$-regular case for max cut and max/min bisection?