The Peculiar Phase Structure of Random Graph Bisection

Allon G. Percus

School of Mathematical Sciences
Claremont Graduate University

September 3, 2009
Collaborators

- Gabriel Istrate
- Bruno Gonçalves
- Robert Sumi
- Stefan Boettcher

Outline

1 Background
   - Phase Structure
   - Graph Bisection Problem

2 Random Graph Bisection
   - Previous Results
   - Upper Bound on Bisection Width
   - Computational Consequences
Consider random 3-SAT, and look at space of all satisfying assignments of a formula.

- Define two solutions to be adjacent if Hamming distance is small: at most $o(n)$ variables differ in value.
- For small $\alpha$, all solutions lie in a single “cluster”: any two solutions are linked by a path of adjacent solutions.
“Usual” Scenario

Pr [satisfiable] vs. Clause–to–variable ratio $\alpha$

- Below a threshold $\alpha_c$: RS, single solution cluster.
- Above $\alpha_c$: RSB, cluster fragments into multiple non-adjacent clusters.

Computational cost
"Usual" Scenario

Below a threshold $\alpha_d < \alpha_c$: RS, single solution cluster.
“Usual” Scenario

Below a threshold $\alpha_d < \alpha_c$: RS, single solution cluster.

Above $\alpha_d$: RSB, cluster fragments into multiple non-adjacent clusters.
“Usual” Scenario

Below a threshold $\alpha_d < \alpha_c$: RS, single solution cluster.

Above $\alpha_d$: RSB, cluster fragments into multiple non-adjacent clusters.
Algorithmic Consequences

- Cluster fragmentation is associated with formation of frozen variables: local backbone of variables that take on same value within a cluster of solutions.

- This traps algorithms: lots of satisfying assignments but hard to find them, making it a “hard satisfiable” subphase.

- But physical picture also motivates new algorithms: survey propagation explicitly takes account of cluster structure, fixing only those variables that are frozen within a cluster.
Definition

- Graph $G = (V, E)$, $|V|$ even
Graph G = (V, E), |V| even

Partition V into two disjoint subsets V₁ and V₂, |V₁| = |V₂|

Minimize bisection width
w = \left| (u, v) \in E : u \in V₁, v \in V₂ \right|:
number of edges with an endpoint in each subset

Applications: computer chip design, resource allocation, image processing
Corresponding **decision** problem is in P: is there a perfect bisection \((w = 0)\)?

**Optimization** problem is NP-hard.

What about random instances \((\mathcal{G}_{np} \text{ ensemble})\)?
Structure of $\mathcal{G}_{np}$ Graphs

Mean degree of graph is $\alpha = p(n - 1)$. The following results on the birth of the giant component are known [Erdős-Rényi, 1959]:

- For $\alpha < 1$, only very small components exist: size $O(\log n)$.

- For $\alpha > 1$, there exists a giant component of expected size $gn$, $g = 1 - e^{-\alpha g}$. All other components: size $O(\log n)$.

- Expected fraction of isolated vertices is $(1 - p)^{n-1} \approx e^{-\alpha}$. 
Mean degree of graph is $\alpha = p(n - 1)$. The following results on the birth of the giant component are known [Erdős-Rényi, 1959]:

- For $\alpha < 1$, only very small components exist: size $O(\log n)$.

- For $\alpha > 1$, there exists a giant component of expected size $gn$, $g = 1 - e^{-\alpha g}$. All other components: size $O(\log n)$.
  - At $\alpha = 2 \log 2$, $g = 1/2$

- Expected fraction of isolated vertices is $(1 - p)^{n-1} \approx e^{-\alpha}$.
  - At $\alpha = 2 \log 2$, $n/4$ isolated vertices
Consequence: Bisection Width

Known results and bounds [Luczak & McDiarmid, 2001]:

- For $\alpha < 1$, $w = 0$ w.h.p.
  - Enough small components to guarantee perfect bisection

- For $1 < \alpha < 2 \log 2$, also $w = 0$ w.h.p.
  - Even close to $\alpha = 2 \log 2$, where the giant component almost occupies entire partition, enough isolated vertices to guarantee perfect bisection

- For $\alpha > 2 \log 2$, $w \propto n$ and obvious upper bound $w/n \leq \alpha/2$ w.h.p.
  - For $2 \log 2 < \alpha < 4 \log 2$, $w/n \leq (\alpha - \log 2)/4$ w.h.p.

[Goldberg & Lynch, 1985] Still leaves a gap at $\alpha = 2 \log 2$. Can we do better?
Consequence: Bisection Width

Known results and bounds [Luczak & McDiarmid, 2001]:

- For $\alpha < 1$, $w = 0$ w.h.p.
  - Enough small components to guarantee perfect bisection
- For $1 < \alpha < 2 \log 2$, also $w = 0$ w.h.p.
  - Even close to $\alpha = 2 \log 2$, where the giant component almost occupies entire partition, enough isolated vertices to guarantee perfect bisection
Consequence: Bisection Width

Known results and bounds [Luczak & McDiarmid, 2001]:

- For $\alpha < 1$, $w = 0$ w.h.p.
  - Enough small components to guarantee perfect bisection
- For $1 < \alpha < 2 \log 2$, also $w = 0$ w.h.p.
  - Even close to $\alpha = 2 \log 2$, where the giant component almost occupies entire partition, enough isolated vertices to guarantee perfect bisection
- For $\alpha > 2 \log 2$, $w = \Omega(n)$ and obvious upper bound $w/n \leq \alpha/2$ w.h.p.
Known results and bounds [Luczak & McDiarmid, 2001]:

- For $\alpha < 1$, $w = 0$ w.h.p.
  - Enough small components to guarantee perfect bisection
- For $1 < \alpha < 2 \log 2$, also $w = 0$ w.h.p.
  - Even close to $\alpha = 2 \log 2$, where the giant component almost occupies entire partition, enough isolated vertices to guarantee perfect bisection
- For $\alpha > 2 \log 2$, $w = \Omega(n)$ and obvious upper bound $w/n \leq \alpha/2$ w.h.p.
- For $2 \log 2 < \alpha < 4 \log 2$, $w/n \leq (\alpha - \log 2)/4$ w.h.p. [Goldberg & Lynch, 1985]
Consequence: Bisection Width

Known results and bounds [Luczak & McDiarmid, 2001]:

- For $\alpha < 1$, $w = 0$ w.h.p.
  - Enough small components to guarantee perfect bisection
- For $1 < \alpha < 2 \log 2$, also $w = 0$ w.h.p.
  - Even close to $\alpha = 2 \log 2$, where the giant component almost occupies entire partition, enough isolated vertices to guarantee perfect bisection
- For $\alpha > 2 \log 2$, $w = \Omega(n)$ and obvious upper bound $w/n \leq \alpha/2$ w.h.p.
- For $2 \log 2 < \alpha < 4 \log 2$, $w/n \leq (\alpha - \log 2)/4$ w.h.p. [Goldberg & Lynch, 1985]

Still leaves a gap at $\alpha = 2 \log 2$. Can we do better?
Consequence: Bisection Width

Experimental results [Boettcher & Percus, 1999]:

- \( w/n \) vs. mean degree

\[ \begin{array}{c|c|c|c|c|c|c|c|c|c} \text{mean degree} & 1.2 & 1.4 & 1.6 & 1.8 & 2 & 2.2 & 2.4 & 2.6 \\ \hline w/n & 0 & 0.02 & 0.04 & 0.06 & 0.08 & 0.1 & \end{array} \]
Consequence: Solution Structure

- For $\alpha < 2\log 2$, all solutions lie in a single cluster (RS) [Istrate, Kasiviswanathan & Percus, 2006]
  - Enough small components that any two solutions are connected by a chain of small swaps preserving balance constraint

- For $\alpha > 2\log 2$, solution space structure is determined by how giant component gets cut
Giant Component Structure

- Giant component consists of a mantle of trees and a remaining core [Pittel, 1990]
- Individual trees are of size $O(\log n)$
- Does optimal cut simply trim trees, or does it slice through core?
Cutting Trees

As long as core is smaller than $n/2$, we can at least get an upper bound on $w$ by restricting cuts to trees.

**Theorem**

Let $\epsilon = \alpha - 2 \log 2$. Then there exists an $\epsilon_0 > 0$ such that for every $\epsilon < \epsilon_0$, w.h.p.

$$\frac{w}{n} < \frac{\epsilon}{\log 1/\epsilon}$$

for graphs with mean degree $\alpha$ in $G_{np}$.

Among other things, this closes the gap at $\alpha = 2 \log 2$. Now how do we prove it?
Cutting Trees

Cut trees starting from largest one until giant component is pruned to size $n/2$:
Cutting Trees

Cut trees starting from largest one until giant component is pruned to size $n/2$:
Cut trees starting from largest one until giant component is pruned to size $n/2$:
Cut trees starting from largest one until giant component is pruned to size $n/2$: 

![Diagram](image-url)
Cutting Trees

Cut trees starting from largest one until giant component is pruned to size $n/2$:
Cut trees starting from largest one until giant component is pruned to size $n/2$: 

![Diagram of trees with a core and mantle]
Let $\delta n$ be “excess” of giant component, $\delta = g - 1/2$.

Let $b_n$ be number of nodes in mantle.

Then $\delta / b$ is fraction of mantle’s nodes to cut.

Now find largest $t_0$ such that $\delta / b$ equals fraction of nodes living on trees of size $\geq t_0$.

If $P(t)$ is distribution of tree sizes on mantle,

$$\frac{\delta}{b} = \frac{\sum_{t=t_0}^{\infty} tP(t)}{\sum_{t=1}^{\infty} tP(t)}$$

The number of trees of size $\geq t_0$ is then

$$w' = \sum_{t=t_0}^{\infty} P(t) \frac{b_n}{\sum_{t=1}^{\infty} tP(t)}$$
Fortunate result of probabilistic independence in $G_{np}$ [Janson et al, 2000]:

- $P(t)$ is simply given by # of ways of constructing tree of size $t$ from $q$ roots ($q = (g - b)n$, size of core) and $r$ other nodes ($r = bn$, size of mantle).
- This is “just combinatorics”:

$$
P(t) = \binom{r}{t} t^t \frac{q}{r} \frac{(q + r - t)^{r-t+1}}{(q + r)^{r-1}}$$

- Let $\rho = b/g$. Then at large $n$,

$$
P(t) \approx \frac{t^t e^{-\rho t}}{t!} \rho^{t-1}(1 - \rho)$$
We now have enough to calculate (or at least bound) $w'$. The rest of the proof is just cleaning up.

That gives the upper bound we need on bisection width $w$.

Theorem implies that $w/n$ scales superlinearly in $\epsilon = \alpha - 2 \log 2$ for small $\epsilon$. This turns out to have physical and algorithmic consequences.

This holds for every $\epsilon < \epsilon_0$, but $\epsilon_0$ may be very small!
Expander Core of Giant Component

Look more closely at giant component structure. Define notion of expander graphs:

- Given graph $G = (V, E)$, imagine cutting $V$ into two subsets $V_1$ and $V_2$ (w.l.o.g. let $|V_1| \leq |V_2|$).

- Expansion of this cut is

$$h = \frac{|(u, v) \in E : u \in V_1, v \in V_2|}{|V_1|},$$

i.e., # of cuts per vertex.

- If in a sequence of graphs of increasing size, expansion of all cuts is bounded below by a constant, these are known as expander graphs.
Giant component is not an expander: cutting the largest tree gives expansion $h \sim 1/\log n$. [Benjamini et al, 2006]. Decorations have certain tree-like properties, and are of size $O(\log n)$. 
Giant component is not an expander: cutting the largest tree gives expansion $h \sim \frac{1}{\log n}$.

But it is a "decorated expander" with an identifiable expander core. [Benjamini et al, 2006].

Decorations have certain tree-like properties, and are of size $O(\log n)$. 
Giant component is not an expander: cutting the largest tree gives expansion $h \sim 1/\log n$.

But it is a “decorated expander” with an identifiable expander core. [Benjamini et al, 2006].
Giant component is not an expander: cutting the largest tree gives expansion $h \sim 1/\log n$.

But it is a "decorated expander" with an identifiable expander core. [Benjamini et al, 2006].
Giant component is not an expander: cutting the largest tree gives expansion $h \sim 1/\log n$.

But it is a "decorated expander" with an identifiable expander core. [Benjamini et al, 2006].
Giant component is not an expander: cutting the largest tree gives expansion $h \sim 1 / \log n$.

But it is a “decorated expander” with an identifiable expander core. [Benjamini et al, 2006].

Decorations have certain tree-like properties, and are of size $O(\log n)$.
Claim

There exists an \( \alpha_d > 2 \log 2 \) such that for all \( \alpha < \alpha_d \), an optimal bisection cannot cut any finite part of the expander core.
Claim

There exists an \( \alpha_d > 2 \log 2 \) such that for all \( \alpha < \alpha_d \), an optimal bisection cannot cut any finite part of the expander core.

Idea:

- Let \( \epsilon = \alpha - 2 \log 2 \). From superlinearity of optimal bisection width, \( w/\epsilon n \to 0 \) as \( \epsilon \to 0 \).
- Number of vertices cut from giant component \( \sim \epsilon n \), so optimal cut requires arbitrarily small expansion.
- Expander core cannot have cuts with vanishing expansion, so for \( \epsilon \) below some constant, optimal cut must avoid expander core.
Apparent Consequences: Solution Structure

- For all $\alpha < \alpha_d$, optimal bisections only cut decorations.
- Since decorations are small, similar arguments seem to apply as for $\alpha < 2 \log 2$: any two optimal bisections are connected by a chain of small swaps preserving balance constraint.
- All solutions then lie in a single cluster (RS) up to $\alpha_d$.
- Suggests that unlike in SAT, $\alpha_d > \alpha_c$ ! This would be first known example where single cluster persists through and beyond critical threshold.
For $\alpha < \alpha_d$, optimal bisection can be found by ranking expansion of decorations.

As in tree-cutting upper bound, cut decorations in increasing order of expansion until giant component is pruned to size $n/2$.

Decorations can be found in polynomial time [Benjamini et al, 2006].

Difficulty is that unlike for trees, it could be best to cut a decoration in the middle.

But decorations are small ($O(\log n)$), and deciding where to cut a given decoration is primarily a bookkeeping operation: takes $2^{O(\log n)} = n^{O(1)}$ operations.
Apparent Consequences: Algorithmic Complexity

**Conjecture**

*For graphs with mean degree $\alpha < \alpha_d$ in $G_{np}$, there exists an algorithm that finds the optimal bisection, w.h.p., in polynomial time.*

If this conjecture holds, it will provide a striking example of an NP-hard problem where typical instances near the phase transitions are not hard.
For graphs in $G_{np}$, new upper bound on bisection width that closes the gap at the critical threshold $\alpha_c$.

All solutions appear to lie in a single cluster (RS) up to and beyond $\alpha_c$, with an RSB transition possibly taking place above this threshold.

Hardest instances do not appear to be concentrated at $\alpha_c$.

Analyzing ensembles of structured random graphs, such as those in $G_{nr}$, remains largely an open problem.