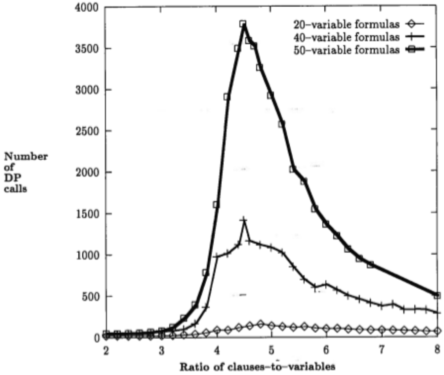
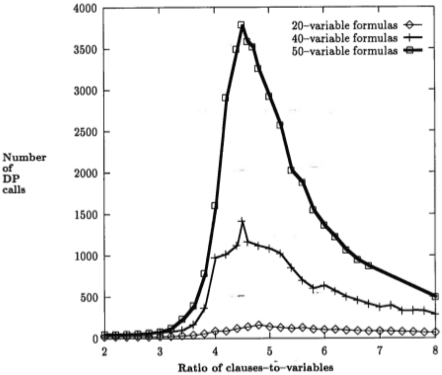
Algorithms for random k-SAT and k-colourings of a random graph

Michael Molloy

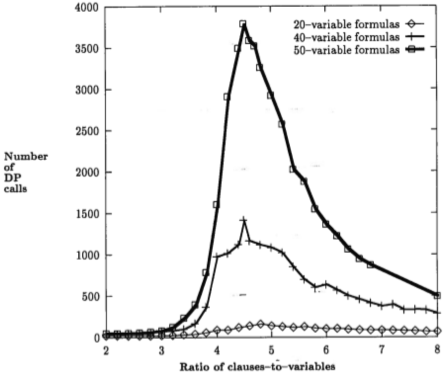
Dept of Computer Science
University of Toronto



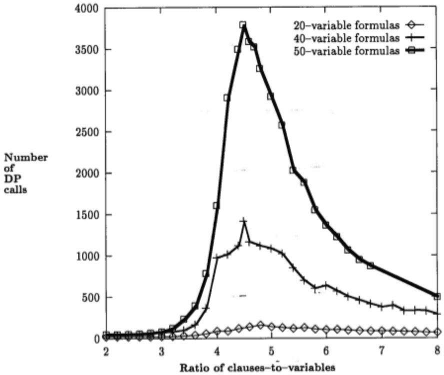
Hard and Easy Distributions of SAT Problems.
Mitchell, Selman, Levesque 1992



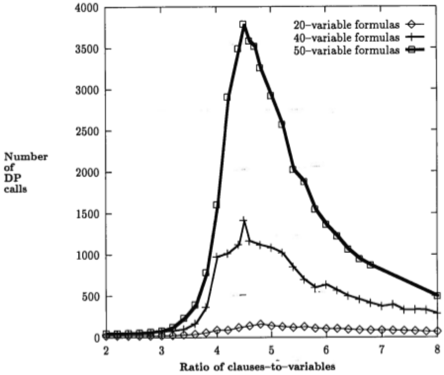
3-SAT: $(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5 \vee \overline{x_7}) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee x_6 \vee \overline{x_7})$



Motivation: Are only a few worse-case k-SAT problems difficult? What about average problems?



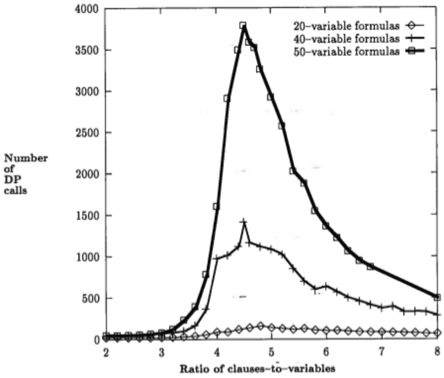
Question: What makes them difficult?



Question: What makes them difficult?

Chvátal, Szemeredi 1988 W.h.p. the resolution complexity is exponentially high.

Implies that any Davis-Putnam type algorithm will require exponential time to recognize an unsatisfiable formula.



Question: What makes them difficult?

Achlioptas, Beame, M 2001 Explains why it takes a long time to recognize a satisfiable formula.

Survey Propogation

Finds satisfying solutions with $n = 1,000,000$ and $M = 4.25n$.

(Satisfiabilty threshold is ≈ 4.267)

Mezard, Zecchina 2002

Survey Propogation

Finds satisfying solutions with $n = 1,000,000$ and $M = 4.25n$.

(Satisfiabilty threshold is ≈ 4.267)

Mezard, Zecchina 2002

Gave us structural properties about the solutions that explain the algorithmic difficulties.

Random Models

Random k-SAT: n variables and $M = rn$ clauses.

Random Models

Random k-SAT: n variables and $M = rn$ clauses.

$G_{n,M}$: Random graph with n vertices and $M = rn$ edges.

Erdős, Rényi 1959

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:

If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:

If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

$$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_2}) \wedge \ldots$$

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:

If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

$$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_2}) \wedge \dots$$

$$x_2 = F$$

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:

If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

$$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_2}) \wedge \dots$$

$$x_2 = F$$

$$(x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge \dots$$

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:
If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

$$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_2}) \wedge \dots$$

$$x_2 = F$$

$$(x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge \dots$$

3-SAT: Works up to density $r < 2.666$; threshold ≈ 4.267
k-SAT: Works up to density $r < \frac{2^k}{k}$; threshold $\approx 2^k \ln 2$
(Franco, Paull 1983; Achlioptas, Peres 2004; Coja-Oghlan 2013)

A Simple Greedy Algorithm

UNIT CLAUSE

Iterate:

If there is a clause of size one, set that variable.
Else pick a random variable and set it randomly.

$$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_2}) \wedge \dots$$

$$x_2 = F$$

$$(x_5) \wedge (x_1 \vee \overline{x_3} \vee \overline{x_5}) \wedge \dots$$

3-SAT: Works up to density $r < 2.666$; threshold ≈ 4.267

k-SAT: Works up to density $r < \frac{2^k}{k}$; threshold $\approx 2^k \ln 2$

Variants of this algorithm all fail to work above $r = O\left(\frac{2^k}{k}\right)$.

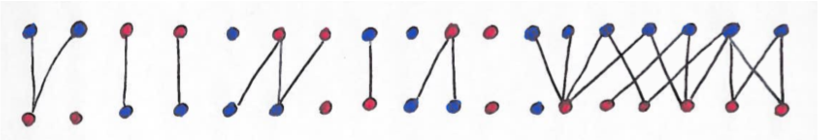
Clustering

Roughly speaking, clusters are:

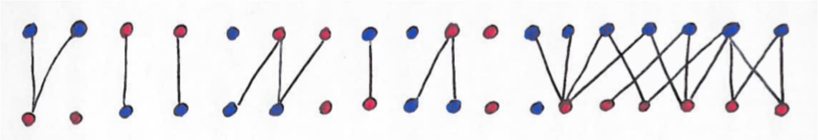
- Well-connected. One can move throughout the cluster changing $o(n)$ vertices at a time.
- Well-separated Moving from one cluster to another requires changing $\Theta(n)$ vertices in one step.

Parisi, Mezard, Zecchina

2-colourings of a Random Bipartite Graph

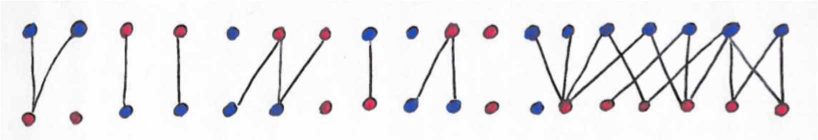


2-colourings of a Random Bipartite Graph



Two clusters - one for each colouring of the giant component.

2-colourings of a Random Bipartite Graph



Two clusters - one for each colouring of the giant component.

We can move within a cluster by switching one small component at a time.

But leaving a cluster requires switching the $\Theta(n)$ vertices in the giant component.

Clustering

k-SAT clusters: $\approx \frac{2^k \ln k}{k}$ unsatisfiable: $\approx 2^k \ln 2$

k-COL clusters: $\approx \frac{1}{2} k \ln k$ unsatisfiable: $\approx k \ln k$

WALK-SAT

Start with any assignment.
While there are unsatisfied clauses:

  Pick a random unsatisfied clause.
  Randomly choose one of its variables and flip it.

$$(x_1 \vee \overline{x_4} \vee x_5) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_5}) \wedge (x_3 \vee \overline{x_4} \vee \overline{x_5}) \wedge \ldots$$

$$x_1 = T, x_2 = T, x_3 = T, x_4 = T, x_5 = T$$

WALK-SAT

Start with any assignment.
While there are unsatisfied clauses:
    Pick a random unsatisfied clause.
    Randomly choose one of its variables and flip it.

$$(x_1 \vee \overline{x_4} \vee x_5) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_5}) \wedge (x_3 \vee \overline{x_4} \vee \overline{x_5}) \wedge \ldots$$

$$x_1 = T, x_2 = T, x_3 = T, x_4 = T, x_5 = T$$

WALK-SAT

Start with any assignment.
While there are unsatisfied clauses:
    Pick a random unsatisfied clause.
    Randomly choose one of its variables and flip it.

$$(x_1 \vee \overline{x_4} \vee x_5) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_5}) \wedge (x_3 \vee \overline{x_4} \vee \overline{x_5}) \wedge \ldots$$

$$x_1 = T, x_2 = T, x_3 = T, x_4 = T, x_5 = T$$

WALK-SAT

Start with any assignment.
While there are unsatisfied clauses:

  Pick a random unsatisfied clause.

  Randomly choose one of its variables and flip it.

$$(x_1 \vee \overline{x_4} \vee x_5) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_5}) \wedge (x_3 \vee \overline{x_4} \vee \overline{x_5}) \wedge \ldots$$

$$x_1 = T, x_2 = T, x_3 = F, x_4 = T, x_5 = T$$

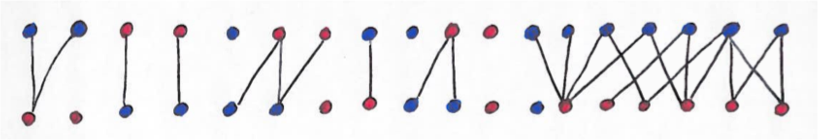
WALK-SAT

Start with any assignment.
While there are unsatisfied clauses:
    Pick a random unsatisfied clause.
    Randomly choose one of its variables and flip it.

Seems to work up to freezing threshold $\approx \frac{2^k \ln k}{k}$

Proven to work up to $\frac{2^k \ln k}{25k}$ (Coja-Oghlan, Frieze 2012)

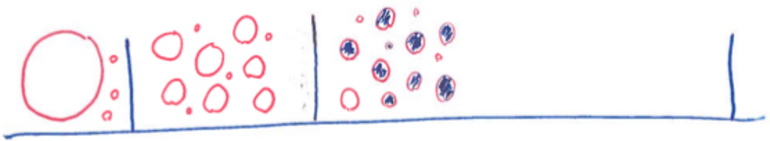
2-colourings of a Random Bipartite Graph



Two clusters - one for each colouring of the giant component.

Every vertex of the giant component is frozen.
Its colour is fixed within each cluster.

The Freezing Threshold



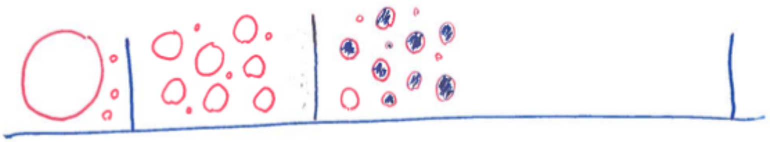
Freezing Threshold ≈ Clustering Threshold

Frozen Variable: Has the same value on every solution in the cluster.

- $r < r^f$: Almost all clusters have no frozen variables.
- $r > r^f$: Almost all clusters have $\Theta(n)$ frozen variables.

Krzakala, Zdeborova; Montanari, Ricci-Tersenghi, Semerjian

The Freezing Threshold

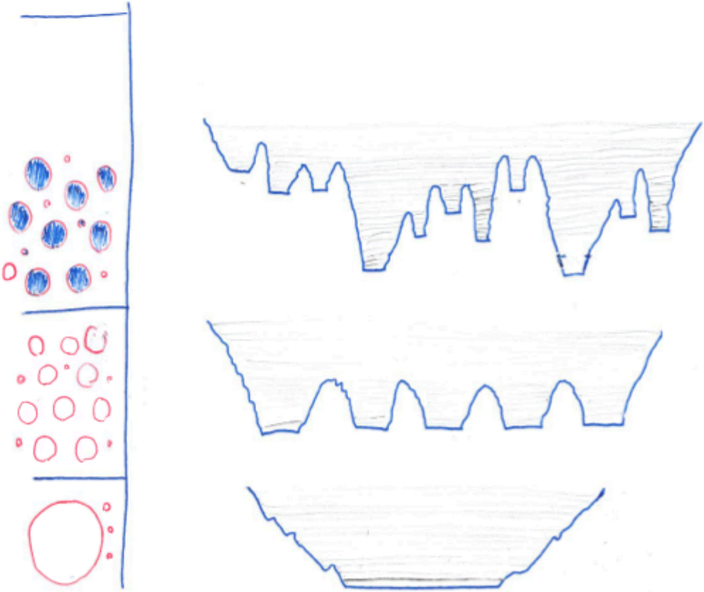


Freezing Threshold ≈ Clustering Threshold

Unfrozen Variable: Can be changed by making a local modficiation - changing $o(n)$ nearby variables.

Frozen Variable: To change it requires a global modification - changing $\Theta(n)$ variables.

- $r < r^f$: Almost all solutions have no frozen variables.
- $r > r^f$: Almost all solutions have $\Theta(n)$ frozen variables.



A Complicated Greedy Algorithm

DECIMATION

Find a variable that is set T (F) in **most** solutions.

Set it T (F).

Iterate.

A Complicated Greedy Algorithm

DECIMATION

Find a variable that is set T (F) in **most** solutions.

Set it T (F).

Iterate.

The marginal of a variable is (p_T, p_F) in a uniformly random solution.

Challenge: Compute the marginals.

Belief Propogation with Decimation

Use BP to estimate the marginal for each variable.

Set the most biased variable.

Iterate.

The marginal of a variable is (p_T, p_F) in a uniformly random solution.

When does BP compute accurate marginals?

BP works perfectly on trees.

When does BP compute accurate marginals?

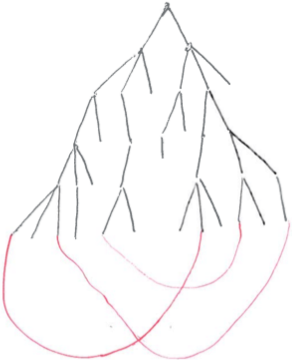
BP works perfectly on trees.

Random graphs look locally like trees.

When does BP compute accurate marginals?

BP works perfectly on trees.

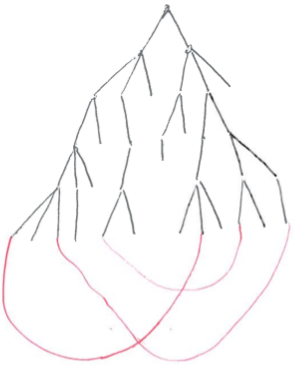
Random graphs look locally like trees.



When does BP compute accurate marginals?

BP works perfectly on trees.

Random graphs look locally like trees.

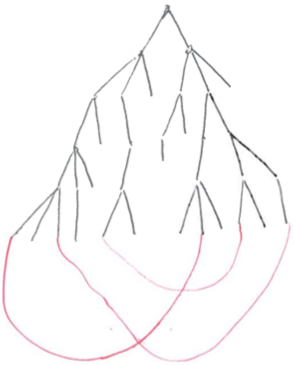


Intuition: BP should be accurate if there is negligible correlation from long paths between leaves.

When does BP compute accurate marginals?

BP works perfectly on trees.

Random graphs look locally like trees.



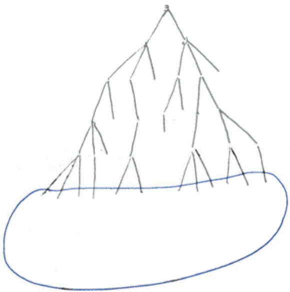
Intuition: BP should be accurate if there is negligible correlation from long paths between leaves.

Equivalently: Two random vertices have negligible correlation.

When does BP compute accurate marginals?

BP works perfectly on trees.

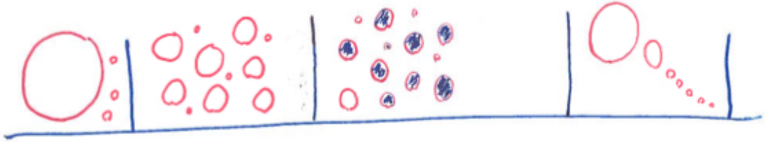
Random graphs look locally like trees.



Intuition: BP should be accurate if there is negligible correlation from long paths between leaves.

Equivalently: Two random vertices have negligible correlation.

Condensation

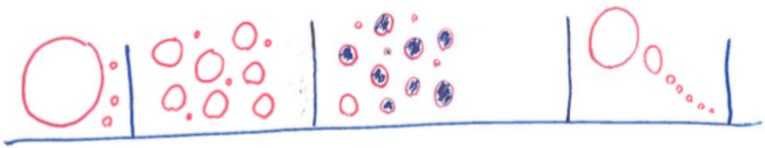


Condensation Threshold ≈ Satisfiability Threshold

Krzakala, Montanari, Ricci-Tersenghi, Semerjian, Zdeborova 2007

After condensation: one cluster contains a linear proportion of the solutions.

Condensation



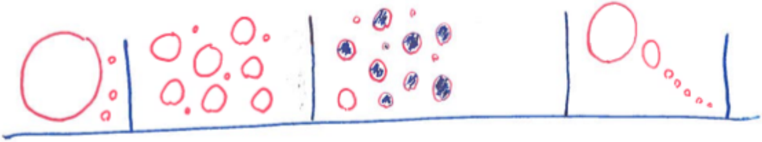
Condensation Threshold ≈ Satisfiability Threshold

Krzakala, Montanari, Ricci-Tersenghi, Semerjian, Zdeborova 2007

After condensation: one cluster contains a linear proportion of the solutions.

This introduces correlations which prevent BP from working.

Condensation



Condensation Threshold $\approx$ Satisfiability Threshold

Krzakala, Montanari, Ricci-Tersenghi, Semerjian, Zdeborova 2007

After condensation: one cluster contains a linear proportion of the solutions.

This introduces correlations which prevent BP from working.

IDEA: Take marginals over random clusters rather than random solutions.

Survey Propogation

In a cluster, a variable can take one of three labels:

- frozen True
- frozen False
- Not frozen

The marginal of a variable is (p_T, p_F, p_*) in a uniformly random cluster.

Survey Propogation

In a cluster, a variable can take one of three labels:

- frozen True
- frozen False
- Not frozen

The marginal of a variable is (p_T, p_F, p_*) in a uniformly random cluster.

The set of valid $\{T, F, *\}$ assignments can be described using local rules.

Eg. if x_i=T then x_i is in a clause where every other literal is False.

Survey Propogation

In a cluster, a variable can take one of three labels:

- frozen True
- frozen False
- Not frozen

The marginal of a variable is (p_T, p_F, p_*) in a uniformly random cluster.

The set of valid $\{T, F, *\}$ assignments can be described using local rules.

Eg. if x_i=T then x_i is in a clause where every other literal is False.

This allows us to use BP to estimate marginals over random $\{T, F, *\}$ assignments.

Survey Propogation with Decimation

Use SP to estimate the marginal for each variable.

Set the variable that is most biased to T or F.

Iterate.

Survey Propogation with Decimation

Use SP to estimate the marginal for each variable.

Set the variable that is most biased to T or F.

Iterate until the marginals are all trivial: $p_* \approx 1$.

Survey Propogation with Decimation

Use SP to estimate the marginal for each variable.

Set the variable that is most biased to T or F.

Iterate until the marginals are all trivial: $p_* \approx 1$.

Then apply WALK-SAT.

When does Survey Propogation fail?

Empirical Observation:

The solutions found by SPD always contain no frozen variables.

When does Survey Propogation fail?

Empirical Observation:
The solutions found by SPD always contain no frozen variables.

Intuition: This is how it reaches a subformula with all trivial marginals.

When does Survey Propogation fail?

Empirical Observation:
The solutions found by SPD always contain no frozen variables.

Intuition: This is how it reaches a subformula with all trivial marginals.

Furthermore: Valid $\{T, F, *\}$ assignments are locked, and we think that this makes them very difficult to find.

When does Survey Propogation fail?

Empirical Observation:
The solutions found by SPD always contain no frozen variables.

Intuition: This is how it reaches a subformula with all trivial marginals.

Furthermore: Valid $\{T, F, *\}$ assignments are locked, and we think that this makes them very difficult to find.

Second Freezing Threshold: Every solution has frozen variables.

The second freezing threshold appears to be a barrier for SPD.

When does Survey Propogation fail?

Empirical Observation:
The solutions found by SPD always contain no frozen variables.

Intuition: This is how it reaches a subformula with all trivial marginals.

Furthermore: Valid $\{T, F, *\}$ assignments are locked, and we think that this makes them very difficult to find.

Second Freezing Threshold: Every solution has frozen variables.

The second freezing threshold appears to be a barrier for SPD.

We think the second freezing threshold is $\approx \frac{2^k \ln k}{k}$.
I.e. it is near the clustering/freezing thresholds.

When does Survey Propogation fail?

Empirical Observation:
The solutions found by SPD always contain no frozen variables.

Intuition: This is how it reaches a subformula with all trivial marginals.

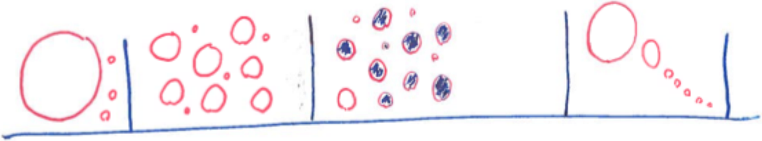
Furthermore: Valid $\{T, F, *\}$ assignments are locked, and we think that this makes them very difficult to find.

Second Freezing Threshold: Every solution has frozen variables.

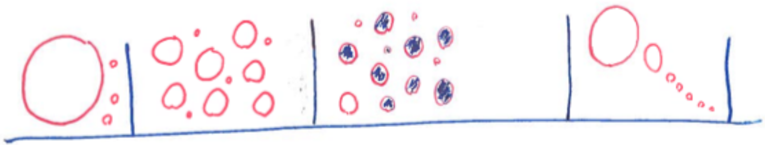
The second freezing threshold appears to be a barrier for SPD.

We think the second freezing threshold is $\approx \frac{2^k \ln k}{k}$.
I.e. it is near the clustering/freezing thresholds.

Proven: It is less than $\frac{4}{5}$ of the satisfiability threshold, for large k.
(Achlioptas, Ricci-Tersenghi 2006).

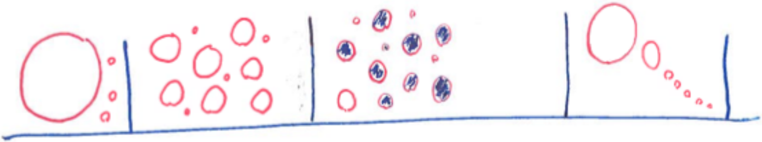


The first freezing threshold is a barrier for WALK-SAT and simple greedy algorithms.



For small k, BPD works a bit past the condensation threshold, and SPD works until the second freezing threshold which is very close to the satisfiability threshold.

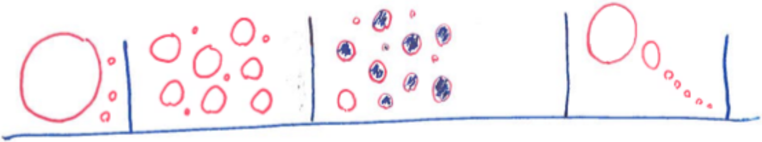
3-SAT: condensation: 3.86, second freezing: 4.25, satisfiability: 4.267



For small k, BPD works a bit past the condensation threshold, and SPD works until the second freezing threshold which is very close to the satisfiability threshold.

3-SAT: condensation: 3.86, second freezing: 4.25, satisfiability: 4.267

For large k, the clustering, first freezing, and second freezing thresholds are all $\approx \frac{2^k \ln k}{k}$, and this seems to be a barrier for BPD and SPD.



For small k, BPD works a bit past the condensation threshold, and SPD works until the second freezing threshold which is very close to the satisfiability threshold.

3-SAT: condensation: 3.86, second freezing: 4.25, satisfiability: 4.267

For large k, the clustering, first freezing, and second freezing thresholds are all $\approx \frac{2^k \ln k}{k}$, and this seems to be a barrier for BPD and SPD.

So despite the early promise of SPD, asymptotically in k it doesn't seem to beat some simple greedy algorithms.

What's Proven?

k-XOR-SAT Each clause has an odd number of true literals.

This is a system of linear equations mod 2.

What's Proven?

k-XOR-SAT Each clause has an odd number of true literals.

This is a system of linear equations mod 2.

The structure of the clusters is much simpler than most CSP's, and it is rigorously very well understood.

Dubois, Mandler 2002
Dietzfelbinger et al 2010
Pittel and Sorkin 2012

Ibrahimi, Kanoria, Kranning, Montanari 2011
Achlioptas, M 2011
Gao, M 2014

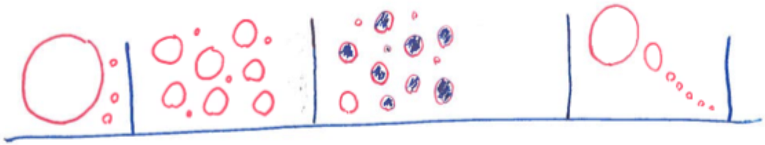
Approximate location of the satisfiability threshold

k-SAT:

$$2^k \ln 2 - (k+1)\frac{\ln 2}{2} - O(1) \le r_{\mathrm{sat}} \le 2^k \ln 2 \qquad \text{Achlioptas, Peres 2004}$$

k-COL:

$$k \ln k - \ln k - O(1) \le r_{\mathrm{sat}} \le k \ln k - \frac{1}{2} \ln k \qquad \text{Achlioptas, Naor 2005}$$



Approximate location of the satisfiability threshold

k-SAT:

$$2^k \ln 2 - (k+1)\frac{\ln 2}{2} - O(1) \leq r_{\rm sat} \leq 2^k \ln 2$$ Achlioptas, Peres 2004

$$r_{\rm sat} = 2^k \ln 2 - \frac{1}{2}(1 + \ln 2) + o(1)$$ Coja-Oghlan 2013

k-COL:

$$k \ln k - \ln k - O(1) \leq r_{\rm sat} \leq k \ln k - \frac{1}{2} \ln k$$ Achlioptas, Naor 2005

$$r_{\rm sat} = k \ln k - \frac{1}{2} \ln k + O(1)$$ Coja-Oghlan, Vilenchik 2013

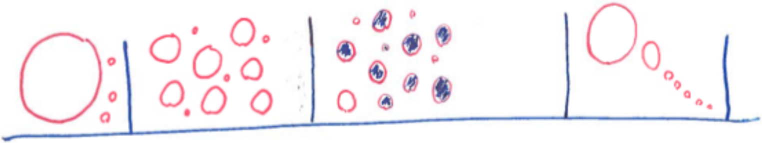
Approximate location of the clustering threshold

k-SAT:

$$r_{\text{cluster}} \leq \frac{2^k \ln k}{k}(1+o(1)) \qquad \text{Achlioptas, Coja-Oghlan 2008}$$

k-COL:

$$r_{\text{cluster}} \leq \frac{1}{2} k \ln k(1+o(1)) \qquad \text{Achlioptas, Coja-Oghlan 2008}$$



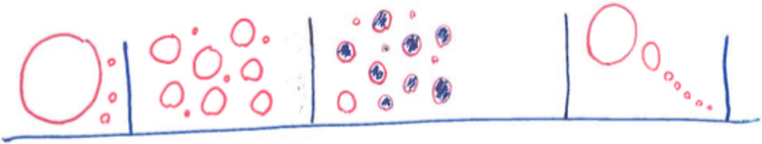
Exact location of the freezing threshold

k-SAT:

$$r_{\text{freeze}} \leq \frac{2^k \ln k}{k}(1 + o(1))$$ Achlioptas, Coja-Oghlan 2008

k-COL:

$$r_{\text{freeze}} = \frac{1}{2} k \ln k(1 + o(1))$$ Achlioptas, Coja-Oghlan 2008



Exact location of the freezing threshold

k-SAT:

$$r_{\text{freeze}} \leq \frac{2^k \ln k}{k}(1 + o(1))$$ Achlioptas, Coja-Oghlan 2008

k-COL:

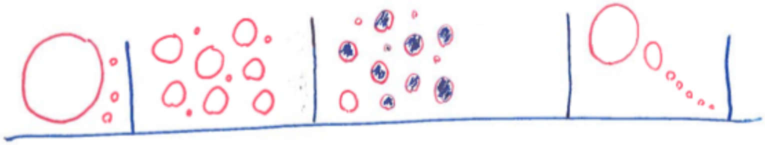
$$r_{\text{freeze}} = \frac{1}{2} k \ln k (1 + o(1))$$ Achlioptas, Coja-Oghlan 2008

$$r_{\text{freeze}} = \min_{x>0} \frac{(k-1)x}{2(1-e^{-x})^{k-1}}$$ M 2012

Exact location of the condensation threshold

k-COL:

The exact value of the condensation threshold is determined.
Bapst, Coja-Oghlan, Hetterich, Rassmann, Vilenchik 2014



Simplified Belief Propogation with Decimation

Use BP to estimate the marginal for each variable.

Pick a random variable and set it randomly according to its marginal.

Iterate.

The marginal of a variable is (p_T, p_F) in a uniformly random solution.

Simplified Belief Propogation with Decimation

Use BP to estimate the marginal for each variable.

Pick a random variable and set it randomly according to its marginal.

Iterate.

The marginal of a variable is (p_T, p_F) in a uniformly random solution.

k-SAT: Fails at densities above $O\left(\frac{2^k}{k}\right)$.

Coja-Oghlan 2011

Simplified Belief Propogation with Decimation

Use BP to estimate the marginal for each variable.

Pick a random variable and set it randomly according to its marginal.

Iterate.

The marginal of a variable is (p_T, p_F) in a uniformly random solution.

k-SAT: Fails at densities above $O\left(\frac{2^k}{k}\right)$.

Coja-Oghlan 2011

After several iterations, the residual formula exhibits condensation and so BP fails.

Local Algorithms fail at densities $>>$ the clustering/freezing threshold.

Gamarnik, Sudan 2013
Rahman, Virag 2014