Solution Clusters for Locked CSP's

Michael Molloy

Dept of Computer Science
University of Toronto

joint with Lenka Zdeborová

Locked Constraint Satisfaction Problems

Introduced by Zdeborová and Mézard 2008

- no constraint has two solutions that differ on a single variable
- each variable lies in at least two constraints

(1-in-k)-SAT
(2-or-5-or-9-in-10)-SAT
XOR-SAT - a.k.a. linear equations mod 2

Locked Constraint Satisfaction Problems

Introduced by Zdeborová and Mézard 2008

- no constraint has two solutions that differ on a single variable
- each variable lies in at least two constraints

(1-in-k)-SAT
(2-or-5-or-9-in-10)-SAT
XOR-SAT - a.k.a. linear equations mod 2

Random instances are extremely hard in the clustered phase.

Roughly 1000 variables can't be solved with Belief Propogation (=SP); BP with reinforcement; Stochastic Local Search.

Locked Constraint Satisfaction Problems

Introduced by Zdeborová and Mézard 2008

- no constraint has two solutions that differ on a single variable
- each variable lies in at least two constraints

(1-in-k)-SAT
(2-or-5-or-9-in-10)-SAT
XOR-SAT - a.k.a. linear equations mod 2

Random instances are extremely hard in the clustered phase.

Roughly 1000 variables can't be solved with Belief Propogation (=SP); BP with reinforcement; Stochastic Local Search.

Note: Changing a variable in a solution will force a cascade of changes, as we must change at least two variables in each affected constraint.

Clusters in Locked Problems

ZM08 studied random locked CSP's with truncated Poisson degree sequences; minimum degree 2.

They found a threshold μ^* such that

- constraint-density below μ^*: all solutions are in a single cluster
- constraint-density above μ^*: every cluster has size $O(1)$.

Clusters in Locked Problems

ZM08 studied random locked CSP's with truncated Poisson degree sequences; minimum degree 2.

They found a threshold μ^* such that

- constraint-density below μ^*: all solutions are in a single cluster
- constraint-density above μ^*: every cluster has size $O(1)$.

Clusters in Locked Problems

ZM08 studied random locked CSP's with truncated Poisson degree sequences; minimum degree 2.

They found a threshold μ^* such that

- constraint-density below μ^*: all solutions are in a single cluster
- constraint-density above μ^*: every cluster has size $O(1)$.

One can travel between any two solutions in the same cluster changing a small number of variables at a time.

To move to a solution in a different cluster requires changing $\Theta(n)$ variables in one step.

Clusters in Locked Problems

ZM08 studied random locked CSP's with truncated Poisson degree sequences; minimum degree 2.

They found a threshold μ^* such that

- constraint-density below μ^*: all solutions are in a single cluster
- constraint-density above μ^*: every cluster has size $O(1)$.

One can travel between any two solutions in the same cluster changing a small number of variables at a time.

To move to a solution in a different cluster requires changing $\Theta(n)$ variables in one step.

We provide a rigorous proof that:

- this is true for XORSAT
- other degree sequences yield many large clusters

Clusters in Locked Problems

ZM08 studied random locked CSP's with truncated Poisson degree sequences; minimum degree 2.

They found a threshold μ^* such that

- constraint-density below μ^*: all solutions are in a single cluster
- constraint-density above μ^*: every cluster has size $O(1)$.

One can travel between any two solutions in the same cluster changing a small number of variables at a time.

To move to a solution in a different cluster requires changing $\Theta(n)$ variables in one step.

We provide a rigorous proof that:

- this is true for XORSAT
- other degree sequences yield many large clusters

EXCEPT: for general degree sequences, we don't have a proof that XORSAT is satisfiable in the clustered phase.

Satisfiability threshold for XORSAT

Random k-XORSAT on a degree sequence with min degree two.

Choose a random k-uniform hypergraph on that degree sequence. Then treat each hyperedge as a clause by signing it randomly.

Random k-XORSAT on a degree sequence with min degree two.

n: number of variables

m: number of edges

Hypothesis

- $m < (1-\epsilon)n$: w.h.p. satisfiable
- $m > (1+\epsilon)n$: w.h.p. unsatisfiable

Satisfiability threshold for XORSAT

Random k-XORSAT on a degree sequence with min degree two.

n: number of variables

m: number of edges

Hypothesis

- $m < (1-\epsilon)n$: w.h.p. satisfiable
- $m > (1+\epsilon)n$: w.h.p. unsatisfiable

Partial Proof: Second moment analysis on number of solutions. The key function f has a local maximum where we want. Missing piece: prove that this is a global maximum.

Satisfiability threshold for XORSAT

Random k-XORSAT on a degree sequence with min degree two.

n: number of variables

m: number of edges

Hypothesis

- $m < (1 - \epsilon)n$: w.h.p. satisfiable
- $m > (1 + \epsilon)n$: w.h.p. unsatisfiable

Partial Proof: Second moment analysis on number of solutions. The key function f has a local maximum where we want. Missing piece: prove that this is a global maximum.

Completed for truncated Poisson degree sequences:

- $k = 3$ Dubois, Mandler 2002
- $k > 3$ Dietzfelbinger et al 2010, Pittel and Sorkin 2012.

Satisfiability threshold for XORSAT

Random k-XORSAT on a degree sequence with min degree two.

n: number of variables

m: number of edges

Hypothesis

- $m < (1-\epsilon)n$: w.h.p. satisfiable
- $m > (1+\epsilon)n$: w.h.p. unsatisfiable

FALSE for some degree sequences. Lelarge 2013

Clusters for k-XORSAT

First studied by Mézard, Ricci-Tersenghi and Zecchina 2003

Found that clustering depended on the 2-core.

Clusters for k-XORSAT

First studied by Mézard, Ricci-Tersenghi and Zecchina 2003

Found that clustering depended on the 2-core.

The subgraph remaining after repeatedly deleting any vertices of degree < 2, and any edges containing those vertices.

Analyzed for random graphs on a fixed degree sequence by Fernholz and Ramachandran 2003

Clusters for k-XORSAT

First studied by Mézard, Ricci-Tersenghi and Zecchina 2003

Found that clustering depended on the 2-core.

- below the 2-core threshold: all solutions in a single cluster
- above the 2-core threshold: there is a cluster for each solution to the 2-core

Clusters for k-XORSAT

First studied by Mézard, Ricci-Tersenghi and Zecchina 2003

Found that clustering depended on the 2-core.

- below the 2-core threshold: all solutions in a single cluster
- above the 2-core threshold: there is a cluster for each solution to the 2-core

Key point:

(Almost) every variable in the 2-core is frozen: changing the value of a variable requires changing $\Theta(n)$ other 2-core variables.

Clusters for k-XORSAT

First studied by Mézard, Ricci-Tersenghi and Zecchina 2003

Found that clustering depended on the 2-core.

- below the 2-core threshold: all solutions in a single cluster
- above the 2-core threshold: there is a cluster for each solution to the 2-core

Key point:

(Almost) every variable in the 2-core is frozen: changing the value of a variable requires changing $\Theta(n)$ other 2-core variables.

Rigorous proofs:

Ibrahimi, Kanoria, Kranning and Montanari (2011)

Achlioptas and M (2011)

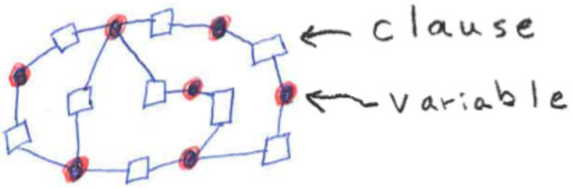
Inside the 2-core threshold: Gao and M (2013)

Why is the 2-core frozen?

Flippable set: A subset S of the variables such that every clause contains an even number of members of S.

Note: Every pair of solutions differs on a flippable set. So

$$\text{“every variable is frozen”} = \text{“every flippable set has linear size”}$$

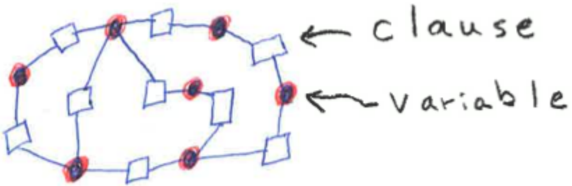


Why is the 2-core frozen?

Flippable set: A subset S of the variables such that every clause contains an even number of members of S.

Note: Every pair of solutions differs on a flippable set. So

"every variable is frozen" $=$ "every flippable set has linear size"



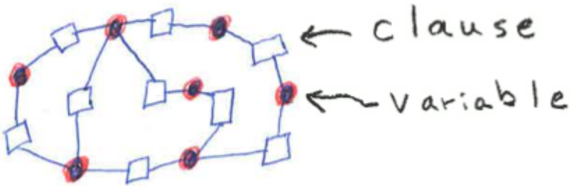
Note: Flippable sets depend only on the underlying hypergraph, not the signs on the clauses, or the actual solution.
So the cluster structure is determined by the hypergraph.

Why is the 2-core frozen?

Flippable set: A subset S of the variables such that every clause contains an even number of members of S.

Note: Every pair of solutions differs on a flippable set. So

"every variable is frozen" $=$ "every flippable set has linear size"



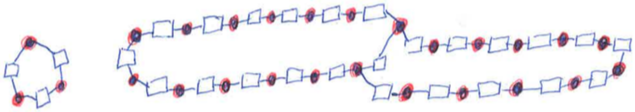
Intuition: If a positive proportion of S has degree greater than 2 in S, then $|S| = \Theta(n)$.

Why is the 2-core frozen?

Flippable set: A subset S of the variables such that every clause contains an even number of members of S.

Intuition: If a positive proportion of S has degree greater than 2 in S, then $|S| = \Theta(n)$.

So small sets must look like:

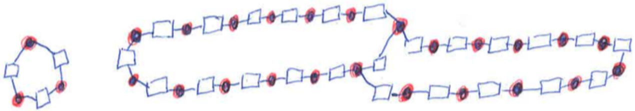


Why is the 2-core frozen?

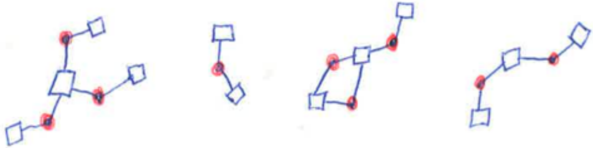
Flippable set: A subset S of the variables such that every clause contains an even number of members of S.

Intuition: If a positive proportion of S has degree greater than 2 in S, then $|S| = \Theta(n)$.

So small sets must look like:



But the subgraph induced by the degree two variables only has small components.



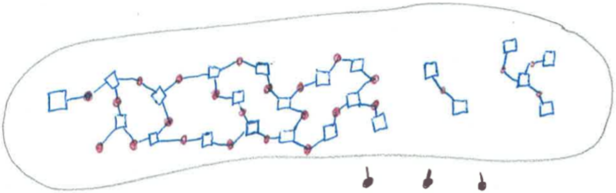
Components in the 2-graph

2-graph: The subgraph of the 2-core induced by the degree 2 variables.

Components in the 2-graph

2-graph: The subgraph of the 2-core induced by the degree 2 variables.

If it had a giant component, then the 2-core would be too fragile.

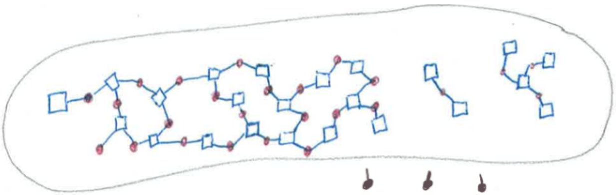


W.h.p. one of the vertices removed near the end of the stripping process would be adjacent to the giant component.
The deletion of that vertex would lead to the deletion of the entire giant component.

Components in the 2-graph

2-graph: The subgraph of the 2-core induced by the degree 2 variables.

If it had a giant component, then the 2-core would be too fragile.



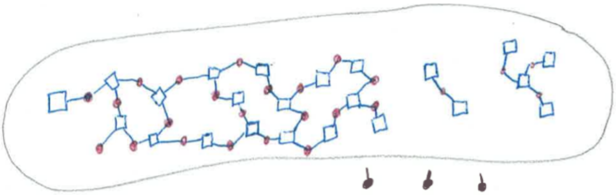
W.h.p. one of the vertices removed near the end of the stripping process would be adjacent to the giant component.
The deletion of that vertex would lead to the deletion of the entire giant component.

Therefore, if a 2-core is reached by a stripping process, then (nearly) every variable in the 2-core is frozen

Components in the 2-graph

2-graph: The subgraph of the 2-core induced by the degree 2 variables.

If it had a giant component, then the 2-core would be too fragile.



However: If we fix an initial degree sequence with minimum degree 2, then we can choose one in which the 2-graph has a giant component.

Degree sequences with minimum degree 2

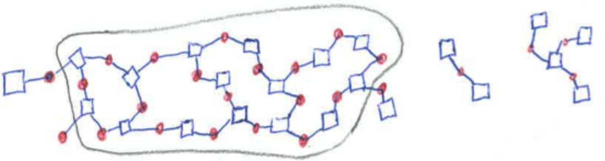
Type 1: The 2-graph has only small components.

Every flippable set has linear size, other than $O(1)$ short cycles.

So every cluster has size $O(1)$.

Type 2: The 2-graph has a giant component

Every vertex in the 2-core of the giant component lies in a flippable cycle of size $O(\log n)$, and so is not frozen.



Degree sequences with minimum degree 2

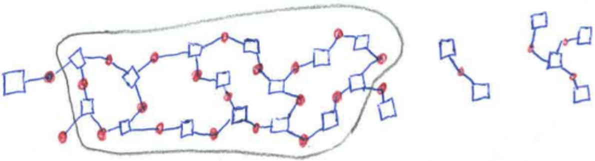
Type 1: The 2-graph has only small components.

Every flippable set has linear size, other than $O(1)$ short cycles.

So every cluster has size $O(1)$.

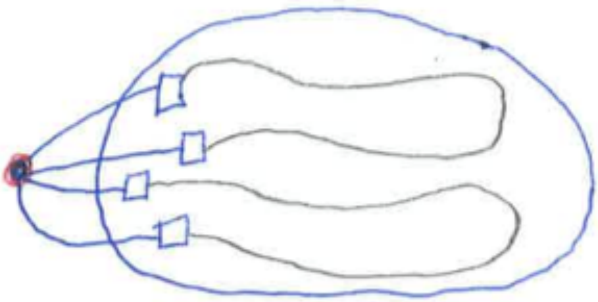
Type 2: The 2-graph has a giant component

Every vertex in the 2-core of the giant component lies in a flippable cycle of size $O(\log n)$, and so is not frozen.

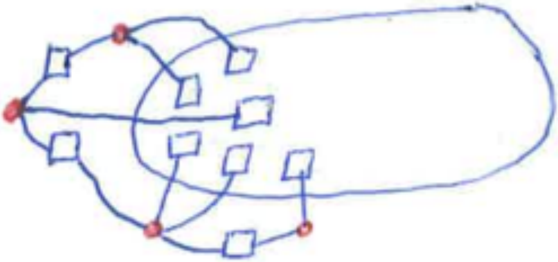


This yields exponentially large clusters, formed by flipping these cycles.

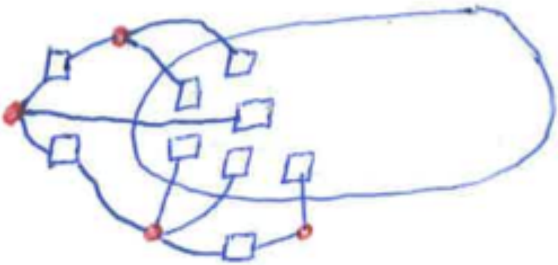
Other flippable sets



Other flippable sets



Other flippable sets



Lemma: *Every vertex lies in one of these flippable sets unless the graph obtained by deleting the giant component of the 2-graph has a 2-core.*

Degree sequences with minimum degree 2

Type 2a: The 2-graph has a giant component. Removing that giant component leaves a graph with no 2-core.

Every vertex in the graph lies in a flippable set of size poly(log n), and so is not frozen.

All solutions lie in a single cluster.

Degree sequences with minimum degree 2

Type 2a: The 2-graph has a giant component. Removing that giant component leaves a graph with no 2-core.

Every vertex in the graph lies in a flippable set of size poly($\log n$), and so is not frozen.

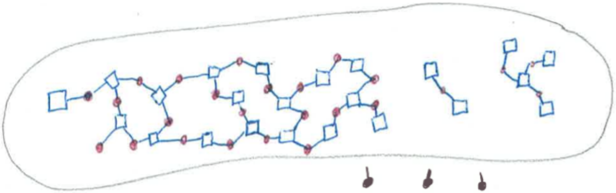
All solutions lie in a single cluster.

Type 2b: The 2-graph has a giant component. Removing that giant component leaves a graph with a 2-core.

Every flippable set in that 2-core has linear size, other than $O(1)$ short cycles.

2-graph: The subgraph of the 2-core induced by the degree 2 variables.

If it had a giant component, then the 2-core would be too fragile.



W.h.p. one of the vertices removed near the end of the stripping process would be adjacent to the giant component.
The deletion of that vertex would lead to the deletion of the entire giant component.

Therefore, if a 2-core is reached by a stripping process, then (nearly) every variable in the 2-core is frozen.

Degree sequences with minimum degree 2

Type 2a: The 2-graph has a giant component. Removing that giant component leaves a graph with no 2-core.

Every vertex in the graph lies in a flippable set of size poly(log n), and so is not frozen.

All solutions lie in a single cluster.

Type 2b: The 2-graph has a giant component. Removing that giant component leaves a graph with a 2-core.

Every flippable set in that 2-core has linear size, other than $O(1)$ short cycles.

Degree sequences with minimum degree 2

Type 2a: The 2-graph has a giant component. Removing that giant component leaves a graph with no 2-core.

Every vertex in the graph lies in a flippable set of size poly($\log n$), and so is not frozen.

All solutions lie in a single cluster.

Type 2b: The 2-graph has a giant component. Removing that giant component leaves a graph with a 2-core.

Every flippable set in that 2-core has linear size, other than $O(1)$ short cycles.

This yields exponentially many clusters, each containing an exponential number of solutions.

Degree sequences with minimum degree 2

Type 2a: The 2-graph has a giant component. Removing that giant component leaves a graph with no 2-core.

Every vertex in the graph lies in a flippable set of size poly($\log n$), and so is not frozen.

All solutions lie in a single cluster.

Type 2b: The 2-graph has a giant component. Removing that giant component leaves a graph with a 2-core.

Every flippable set in that 2-core has linear size, other than $O(1)$ short cycles.

This yields exponentially many clusters, each containing an exponential number of solutions.

Lemma: *Truncated Poisson sequences are Type 1 or Type 2a.*

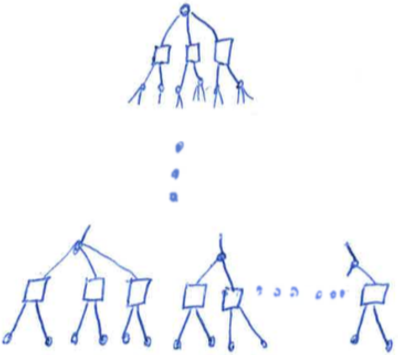
This confirms the cluster description from Zdeborová and Mézard.

Noisy Reconstruction

Choose a Galton-Watson hypertree of height h. Sign the hyperedges to be XORSAT constraints.

Take a random solution, and fix the values of the leaves.

What is the probability that the root is determined (as $h \to \infty$)?



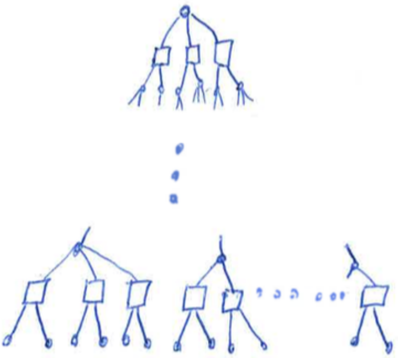
Noisy Reconstruction

Choose a Galton-Watson hypertree of height h. Sign the hyperedges to be XORSAT constraints.

Take a random solution, and fix the values of the leaves.

Erase an ϵ proportion of the leaves.

What is the probability that the root is determined (as $h \to \infty$)?



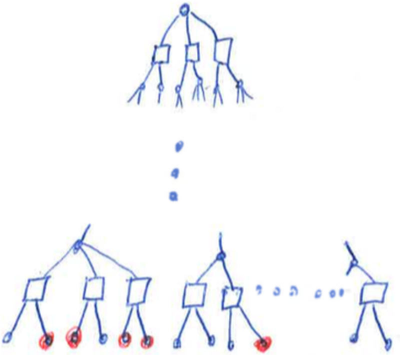
Noisy Reconstruction

Choose a Galton-Watson hypertree of height h. Sign the hyperedges to be XORSAT constraints.

Take a random solution, and fix the values of the leaves.

Erase an ϵ proportion of the leaves.

What is the probability that the root is determined (as $h \to \infty$)?



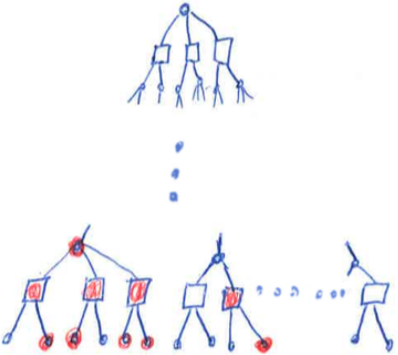
Noisy Reconstruction

Choose a Galton-Watson hypertree of height h. Sign the hyperedges to be XORSAT constraints.

Take a random solution, and fix the values of the leaves.

Erase an ϵ proportion of the leaves.

What is the probability that the root is determined (as $h \to \infty$)?



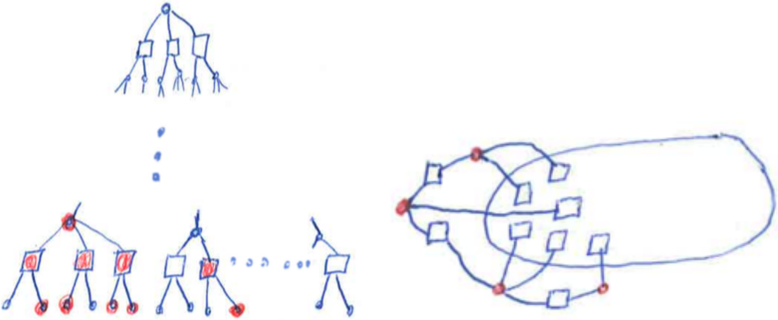
Noisy Reconstruction

Choose a Galton-Watson hypertree of height h. Sign the hyperedges to be XORSAT constraints.

Take a random solution, and fix the values of the leaves.

Erase an ϵ proportion of the leaves.

What is the probability that the root is determined (as $h \to \infty$)?



Noisy Reconstruction

What is the probability that the root is determined (as $h \to \infty$)?

Type 1: 1
Type 2a: 0
Type 2b: $0 < p < 1$
The probability is determined by a fixed point of:

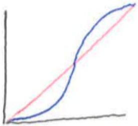
$$f(q) = \frac{\sum i\lambda_i(1-(1-q^{k-1}))^{i-1}}{\sum i\lambda_i}$$

Noisy Reconstruction

What is the probability that the root is determined (as $h \to \infty$)?

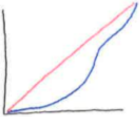
Type 1: 1
Type 2a: 0
Type 2b: $0 < p < 1$
The probability is determined by a fixed point of:

$$f(q) = \frac{\sum i\lambda_i (1 - (1 - q^{k-1}))^{i-1}}{\sum i\lambda_i}$$



Type 1

Type 2b



Type 2a

Other Locked Problems

These techniques should yield similar results regarding flippable sets in other locked problems on the planted model.

Other Locked Problems

These techniques should yield similar results regarding flippable sets in other locked problems on the planted model.

For many balanced problems, eg (2-or-4-in-6)-SAT, the planted model should be contiguous to the standard model, but that requires extensive second moment analysis.

Other Locked Problems

These techniques should yield similar results regarding flippable sets in other locked problems on the planted model.

For many balanced problems, eg (2-or-4-in-6)-SAT, the planted model should be contiguous to the standard model, but that requires extensive second moment analysis.

This will imply, eg. that there are a large number of clusters. But we used the linear algebra structure of XORSAT to prove that there is a single cluster.