Counting Solution Clusters in Graph Coloring Problems Using Belief Propagation

Lukas Kroc, Ashish Sabharwal, Bart Selman — Cornell University

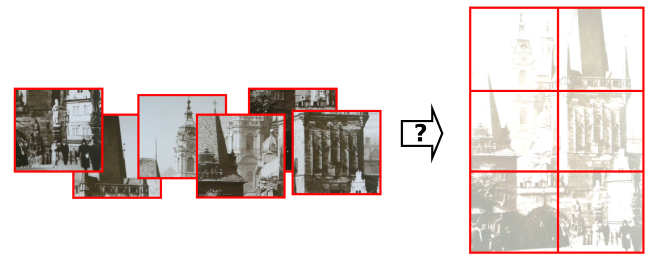
Poster ID: **M48**

1) The Problem

Constraint Satisfaction Problem (CSP)

- Constraint Satisfaction Problem **P**:
  Input: a set V of variables
  a set of corresponding domains of variable values [discrete, finite]
  a set of constraints on V [constraint = set of allowed value tuples]
  Output: a solution, valuation of variables that satisfies all constraints

- Example: a jigsaw puzzle
  - Squares = variables
  - Pieces = domain
  - Matching edges = constraints
  - Full picture = solution



Well-known CSPs:

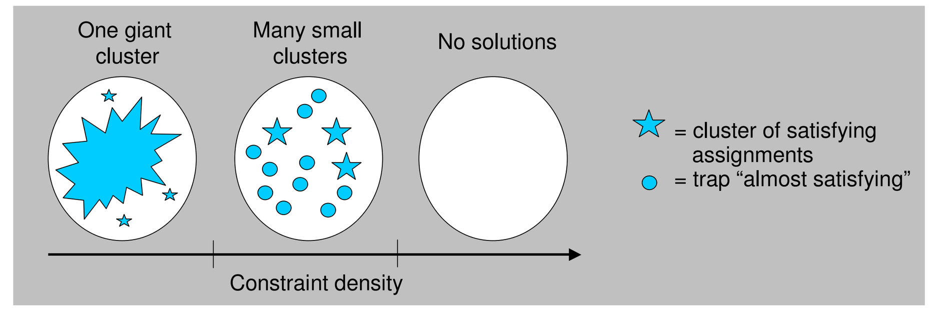
- **k-SAT**: Boolean satisfiability
  - Domains: {0,1} or {true, false}
  - Constraints: disjunctions of variables or their negations ("clauses") with exactly *k* variables each
  - NP complete for k≥3

$$F=\underbrace{(\neg x\vee y\vee z)}_{\alpha}\wedge\underbrace{(x\vee\neg y\vee z)}_{\beta}\wedge\underbrace{(x\vee y\vee\neg z)}_{\gamma}$$

- **k-COL**: Graph coloring
  - Variables: nodes of a given graph
  - Domains: colors 1…*k*
  - Constraints: no two adjacent nodes get the same color.
  - NP complete for k≥3

Solution Space Fragmentation:

- Solution space of random combinatorial problems *fractures into clusters* as constraint density (& hardness) increases



- The fastest solution technique relies on *marginal probability estimates* over clusters (*not* solutions): Survey Propagation (SP) [Mezard et al.'02]

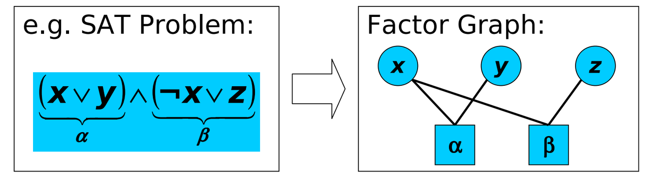
The Quest(ion):

Can solution clusters be reasoned about efficiently?
E.g. can we count them and estimate cluster marginal probabilities

2) Estimating Marginal Probabilities

Factor Graphs:

- Cast the problem as an inference problem in a graphical model, e.g. a factor graph:
  - A bipartite undirected graph with two types of nodes:
    - **Variables**: one node per variable
    - **Factors**: one node per constraint
  - Factor nodes are connected to exactly variables from represented constraint



- Semantics of a factor graph:
  - Each variable node has an associated discrete domain
  - Each factor node α has an associated factor function $f_\alpha(x_\alpha)$, weighting the variable setting. For CSP, it =1 iff associated constraint is satisfied, else =0
  - Weight of the full configuration ***x***:
  - Summing weights of all configurations defines **partition function**:

$$\boxed{Z=\sum_{\mathbf{x}}\prod_\alpha f_\alpha(\mathbf{x}_\alpha)}$$

  - For CSPs the partition function computes the number of solutions

Querying Factor Graphs:

- **What is the value of the partition function *Z*?**
  - E.g. count number of solutions in CSP

$$Z=\sum_{\mathbf{x}}\underbrace{\prod_\alpha f_\alpha(\mathbf{x}_\alpha)}_{=F(\mathbf{x})}$$

- **What are the marginals of the variables?**
  - E.g. fraction of solutions in which a variable *i* is fixed to x_i

$$p_i(x_i)=\frac{1}{Z}\sum_{\mathbf{x}_{-i}}\prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$

Notation: $\mathbf{x}_{-i}$ are all variables except x_i

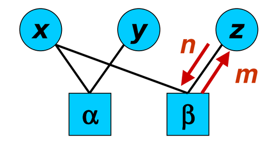
- **What is the configuration with maximum weight *F(x)*?**
  - E.g. finds one (some) solution to a CSP
  - Maximum Likelihood (ML) or Maximum A'Posteriori (MAP) inference

$$\mathrm{argmax}_{\mathbf{x}}\prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$

Belief Propagation:

- Use **Belief Propagation** to do approximate inference:
  - An algorithm in which an "agreement" is to be reached by sending "messages" along edges of the factor graph (Message Passing algorithm)
    - **PROS** — very scalable
    - **CONS** — finicky: exact on tree factor graphs; gives results of uncertain quality on loopy graphs
  - Blackbox BP (for factor graphs):
    - Iteratively solve the following set of recursive equations in [0,1]

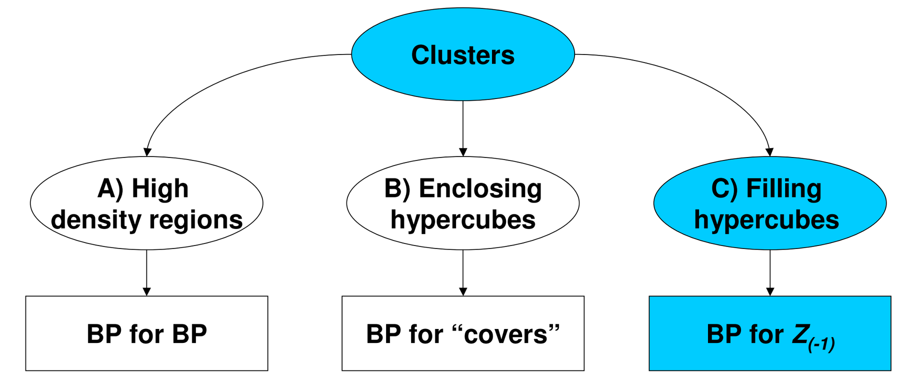
$$n_{i\to\alpha}(x_i)\propto\prod_{\beta\ni i\setminus\alpha}m_{\beta\to i}(x_i)$$
$$m_{\alpha\to i}(x_i)\propto\sum_{\mathbf{x}_{\alpha\setminus i}}f_\alpha(\mathbf{x}_\alpha)\prod_{j\in\alpha\setminus i}n_{j\to\alpha}(x_j)$$



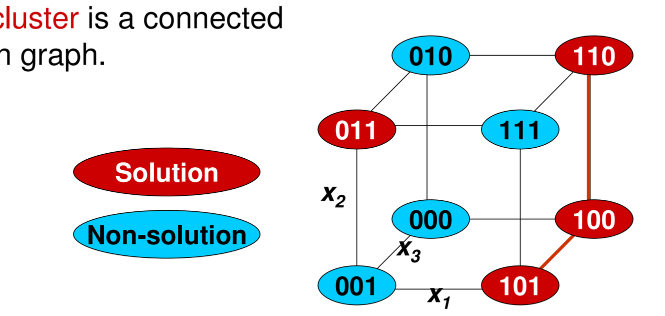
  - **Marginal estimates** (**beliefs** $b_i(x_i)$) (estimates marginal probabilities) and **estimate of number of solutions** Z can be easily computed from fixed point.

3) Clusters

Representing clusters in a factor graph:



- **Definition:** A solution graph is an undirected graph where nodes correspond to solutions and are neighbors if they differ in value of only one variable.
- **Definition:** A solution cluster is a connected component of a solution graph.



Previous Approaches:

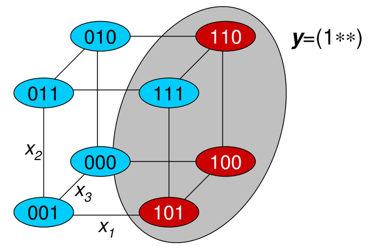
A) High density regions:

- The original SP derivation from statistical mechanics [Mezard et al.'02, Mezard et al.'07]
- A cluster corresponds to a fixed point of BP over the set of all solutions (a "blob" of solutions). Does not work with the definition of a cluster from above.
- BP for clusters is "BP over fixed points of BP" = Survey Propagation

B) Enclosing hypercubes:

- First rigorous derivation of SP for SAT [Braunstein et al. '04, Maneva et al. '05]
- Leads to the concept of "covers"

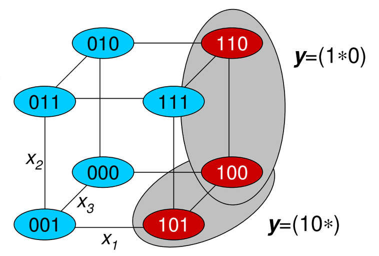
The (unique) minimal hypercube ***y*** enclosing the whole cluster.
1) No solution sticks out: setting any x_i to a value not in y_i cannot be extended to a solution from the cluster.
2) The enclosing is tight: setting any variable x_i to any value from y_i can be extended to a full solution from the cluster.



Our approach:

C) Filling hypercubes

A (non-unique) maximal hypercube fitting entirely inside the cluster.
1) The hypercube ***y*** fits inside the cluster.
2) The hypercube ***y*** cannot grow: extending the hypercube in any direction *i* sticks out of the cluster.



- Allows for provable results for exact cluster counting, as well as new BP style algorithms for efficient estimates.

4) Belief Propagation for Clusters

Factor Graph for Clusters

- To reason about clusters, we seek a factor graph representation
  - Because we can do approximate inference on factor graphs
  - Need to count number of solutions first, with an expression similar to *Z* for solutions:

$$Z=\sum_{\mathbf{x}\in\{0,1\}^n}\underbrace{\prod_\alpha f_\alpha(\mathbf{x}_\alpha)}_{=F(\mathbf{x})}$$

$=F(\mathbf{x})$ = 1 iff ***x*** is a solution

- Indeed, we derive the following for approximating number of clusters:

$$\boxed{Z_{(-1)}=\sum_{\mathbf{y}\in\{0,1,*\}^n}(-1)^{\#_*(\mathbf{y})}\prod_\alpha f'_\alpha(\mathbf{y}_\alpha)}$$

$$f'_\alpha(\mathbf{y}_\alpha)=\prod_{\mathbf{x}_\alpha\in\mathbf{y}_\alpha}f_\alpha(\mathbf{x}_\alpha)$$

Checks whether all points in y_α are good (this is efficient).

  - Syntactically very similar to standard *Z*, which computes exactly number of solutions
  - Exactly counts clusters under certain conditions, as discussed later
  - Analogous expression can be derived also for non-boolean domain

Deriving Belief Propagation for $Z_{(-1)}$

- **$Z_{(-1)}$ can also be approximated with "BP"**: the factor graph remains the same, only the semantics is generalized:
  - Variables: $\mathbf{y}\in\{0,1,*\}^n$
  - Factors: $f'_\alpha(\mathbf{y}_\alpha)$

- And we need to adapt the BP equations to cope with (-1).
- Standard BP equations can be derived as stationary point conditions for continuous constrained optimization problem [Yedidia et al. '05]
  - Let *p(**x**)* be the uniform distribution over solutions of a problem
  - Let *b(**x**)* be a unknown parameterized distribution from a certain family
  - The goal is to **minimize $D_{KL}(b||p)$** over parameters of *b(.)*
  - Use *b(.)* to approximate answers about *p(.)*
- The BP adaptation for $Z_{(-1)}$ follows exactly the same path, and generalizes where necessary.

- We call this adaptation $BP_{(-1)}$:
  - The $BP_{(-1)}$ iterative equations: (The black part is BP)

$$n_{i\to\alpha}(y_i)\propto\prod_{\beta\ni i\setminus\alpha}m_{\beta\to i}(y_i)$$

$$m_{\alpha\to i}(y_i)\propto\sum_{\mathbf{y}_{\alpha\setminus i}\in\{0,1,*\}^{|\alpha|-1}}f'_\alpha(\mathbf{y}_\alpha)\prod_{j\in\alpha\setminus i}(-1)^{\delta(y_j=*)}n_{j\to\alpha}(y_j)$$

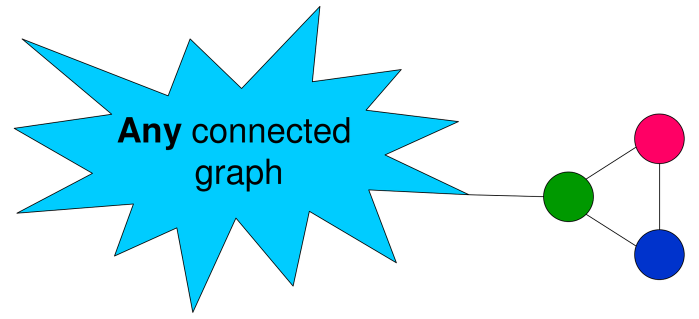
- Relation to Survey Propagation:
  - **For SAT: $BP_{(-1)}$ is equivalent to SP**
    - The $BP_{(-1)}$ equations can be rewritten as SP equations
  - **For COL: $BP_{(-1)}$ is different from SP**
    - $BP_{(-1)}$ estimates the total number of clusters
    - SP estimates the number of clusters with most frequent size

5) Results

Theoretical Results: Exactness of $Z_{(-1)}$

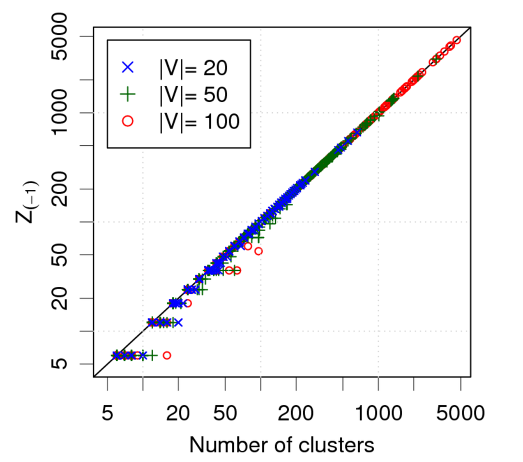
On what kind of solution spaces does $Z_{(-1)}$ count clusters exactly?

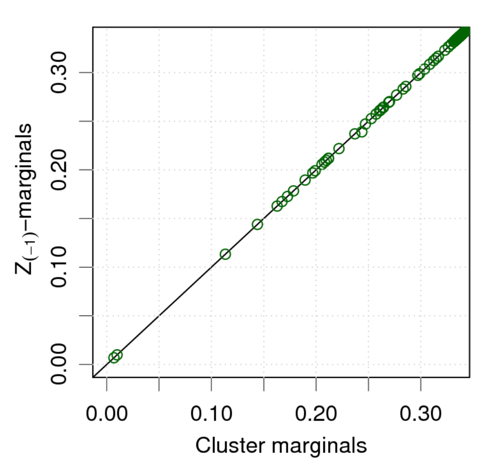
- **Theorem**: $Z_{(-1)}$ is exact for any 2-SAT problem.
- **Theorem**: $Z_{(-1)}$ is exact for a 3-COL problem on *G*, if every connected component of *G* has at least one triangle.



- **Theorem**: $Z_{(-1)}$ is exact if the solution space decomposes into "recursively-monotone subspaces".

Empirical Results: $Z_{(-1)}$ for COL





Empirical Results: $BP_{(-1)}$ for COL

Experiment: approximating $Z_{(-1)}$
1. count exact $Z_{(-1)}$ for many small graphs with avg.deg.∈[1,4.7] (x-axis)
2. compare with $BP_{(-1)}$'s estimate of partition function $Z_{(-1)}$ (y-axis)

Experiment: rescaling # clusters and $Z_{(-1)}$
1. for graphs with various average degrees (x-axis)
2. count $\log(Z_{(-1)})/N$ and $\log(Z_{BP(-1)})/N$ (y-axis)

The rescaling assumes that #clusters=exp(N Σ(c)) Σ(c) is so called **complexity**