Survey Propagation:

Constraint Satisfaction via Message Passing

Constantine Caramanis

Outline

1. Constraint satisfaction problems.
2. Physical intuition: Phase transitions.
3. Graphical model formalism.
4. Message passing algorithms:
    - From Belief Propagation to Survey Propagation
    - From Warning Propagation to Survey Propagation.
5. Open Questions

Constraint Satisfaction

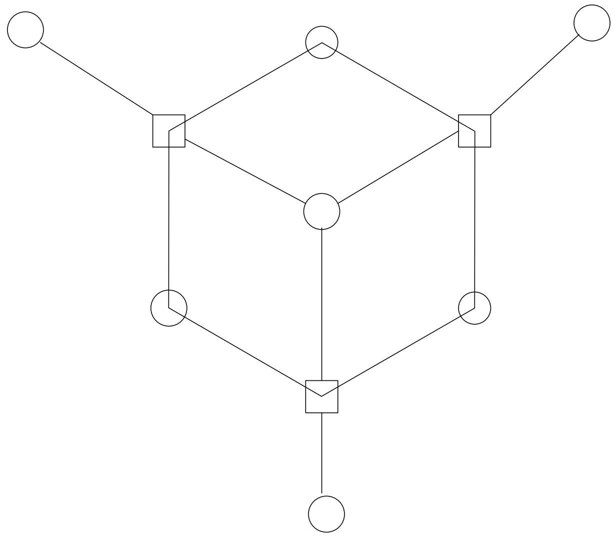
1. Decoding on graphs,

2. Graph coloring problems,

3. k-SAT problem.

Examples I

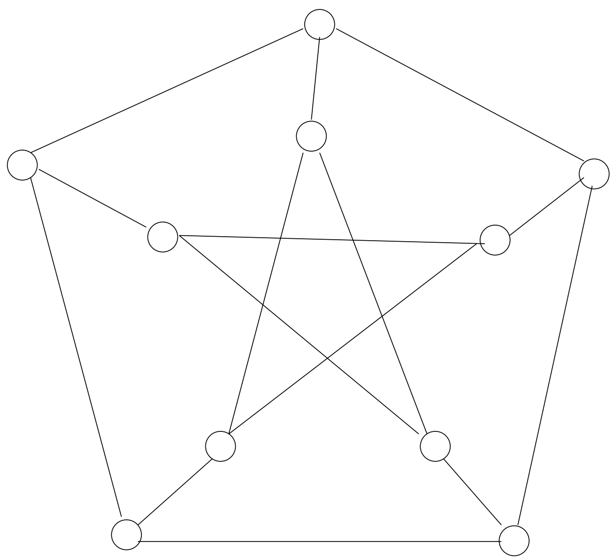
- Coding Problem: Given a received (corrupted) codeword v, find codewrd w that minimizes $d_H(v, w)$. We have a graphical representation:



Examples III

- Graph Coloring: Problem is to color a given graph with some specified number k of colors.

- For example: Can this graph be 4-colored?



Examples III

- The (in)famous 3-SAT problem: Given a conjunction of clauses each containing a disjunction of 3 variables, find a satisfying assignment. For example:

1. $$(x_1 \vee x_4 \vee \tilde{x}_5) \wedge (\tilde{x}_2 \vee \tilde{x}_3 \vee \tilde{x}_4) \wedge (\tilde{x}_1 \vee \tilde{x}_4 \vee x_3) \wedge (\tilde{x}_3 \vee \tilde{x}_4 \vee \tilde{x}_5) \wedge (\tilde{x}_1 \vee x_4 \vee x_2) \wedge (\tilde{x}_1 \vee \tilde{x}_2 \vee x_3)$$

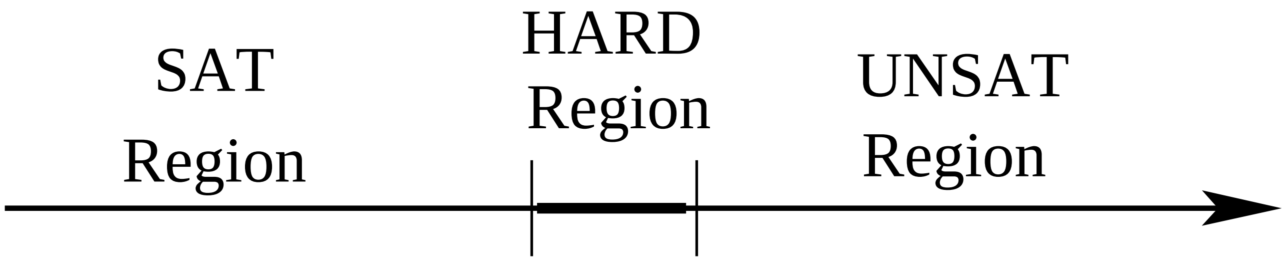
2. $$(x_1 \vee \tilde{x}_2 \vee x_3) \wedge (\tilde{x}_3 \vee \tilde{x}_4 \vee x_5) \wedge (x_5 \vee \tilde{x}_6 \vee \tilde{x}_7) \wedge (x_7 \vee x_8 \vee x_9) \wedge (\tilde{x}_9 \vee x_{10} \vee x_{11}) \wedge (\tilde{x}_{11} \vee \tilde{x}_{12} \vee \tilde{x}_{13}).$$

Random k-SAT

1. Example shows that ratio of clauses to variables is important.

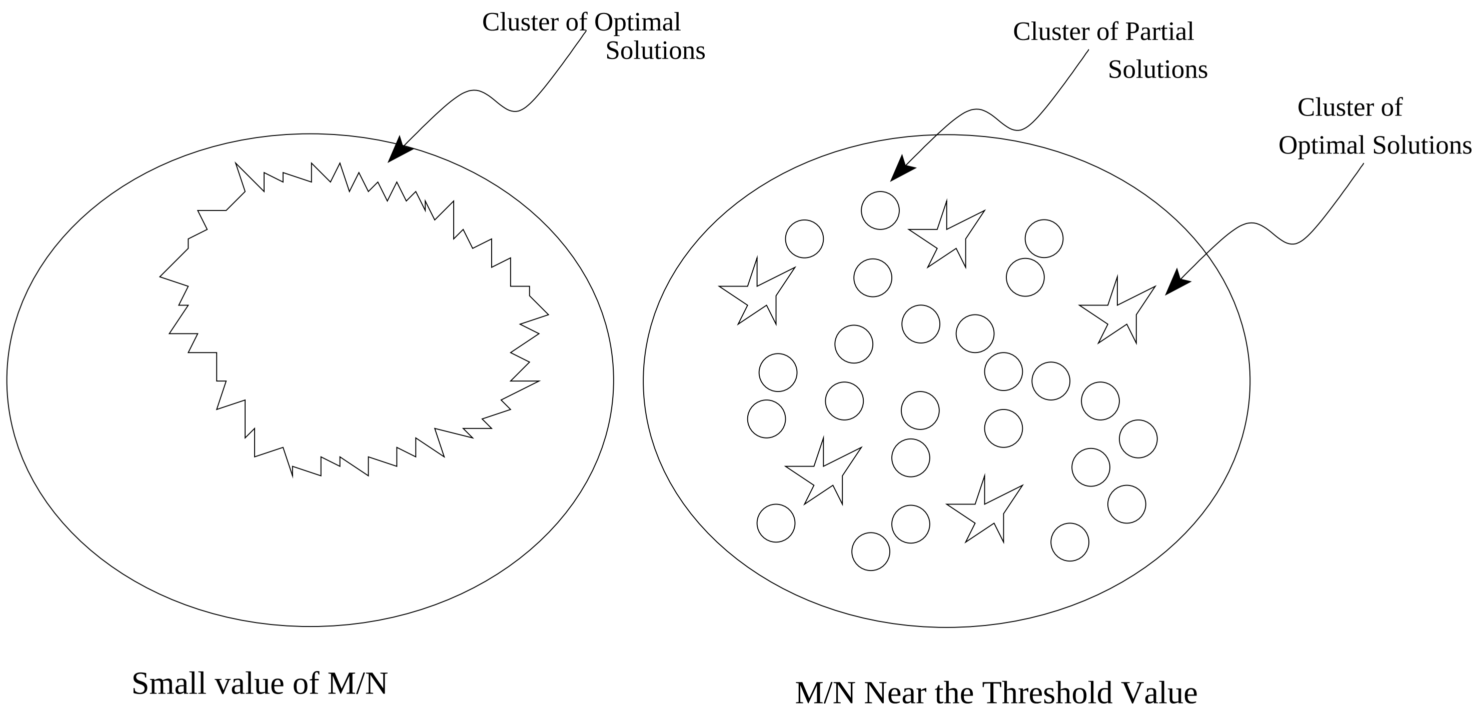
2. N variables, M clauses, fix $\alpha = M/N$.

3. Phase transition: $\alpha_c \approx 4.27$



Physical Intuition

- Clustering effect in solutions space:



Clustering

1. Many solution clusters, with large hamming distance,

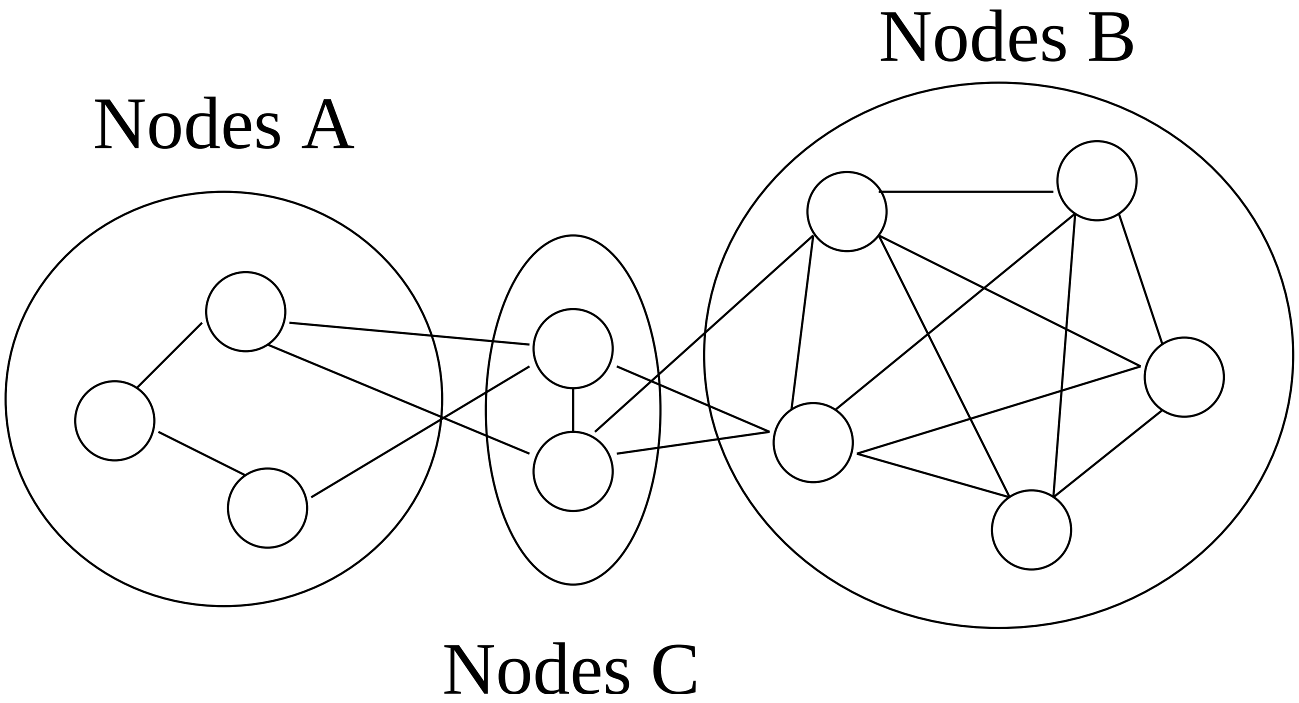
2. Non-convexity: Analogue for constraint satisfaction:

   Partial Solution Clusters

3. Each cluster has frozen and free variables.

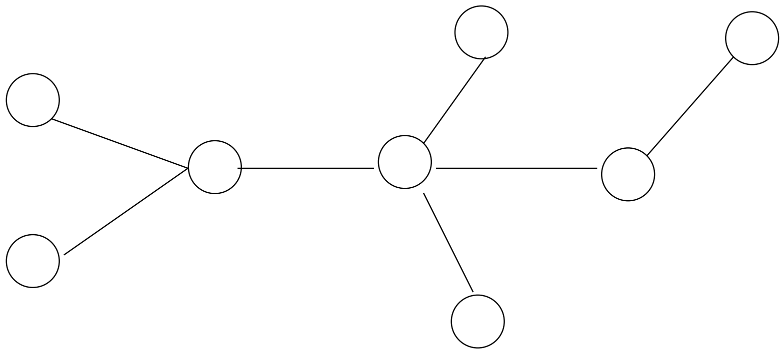
Graphical Models I

- Model conditional independence:



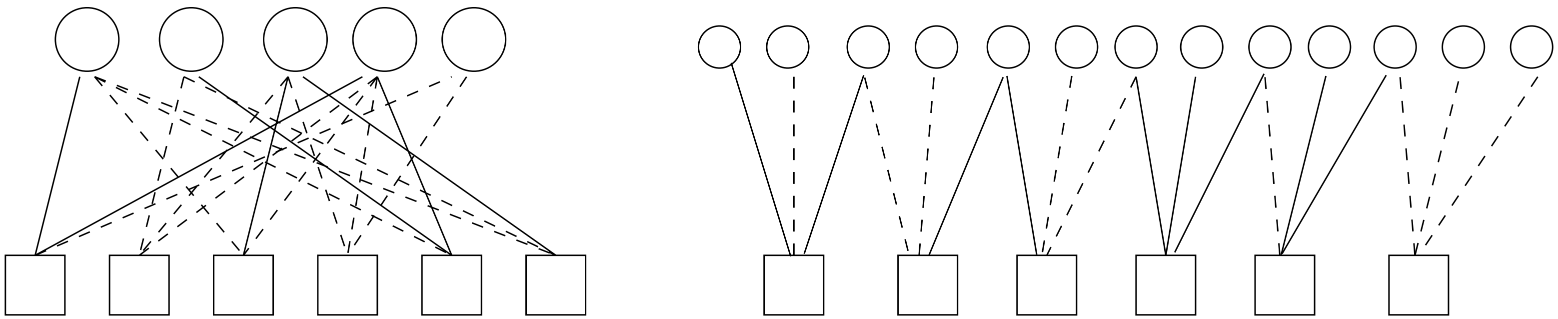
Graphical Models II

1. Tree-structured graphs make life easier: Dynamic Programming



Graphical Models III

- Recall our SAT examples from before.



1. $(x_1 \vee x_4 \vee \tilde{x}_5) \wedge (\tilde{x}_2 \vee \tilde{x}_3 \vee \tilde{x}_4) \wedge (\tilde{x}_1 \vee \tilde{x}_4 \vee x_3) \wedge (\tilde{x}_3 \vee \tilde{x}_4 \vee \tilde{x}_5)$
$\wedge(\tilde{x}_1 \vee x_4 \vee x_2) \wedge (\tilde{x}_1 \vee \tilde{x}_2 \vee x_3)$
2. $(x_1 \vee \tilde{x}_2 \vee x_3) \wedge (\tilde{x}_3 \vee \tilde{x}_4 \vee x_5) \wedge (x_5 \vee \tilde{x}_6 \vee \tilde{x}_7) \wedge (x_7 \vee x_8 \vee x_9)$
$\wedge(\tilde{x}_9 \vee x_{10} \vee x_{11}) \wedge (\tilde{x}_{11} \vee \tilde{x}_{12} \vee \tilde{x}_{13})$.

Message Passing Algorithms I

1. Idea: Nodes pass messages to each other along edges.

2. Key property: Inherently local.

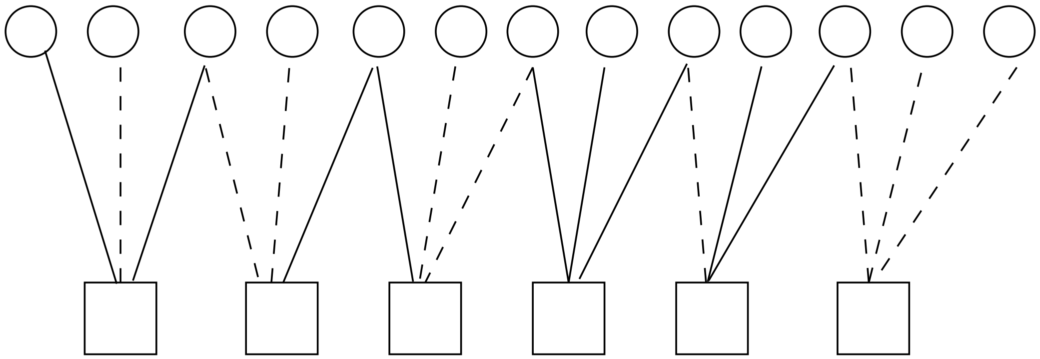
3. Advantage: Efficient computationally.

4. Disadvantage: Assume incoming messages independent.

5. Upshot: Trees are easy, Cycles are hard.

Message Passing Algorithms II

- Tree-structured 3-SAT Example from above:



- Start with: Assignment $X = (0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1)$.

Belief Propagation I

- Variable nodes to function nodes:

$$\mu_{i \to a} = \mathrm{Pr}(\text{variable } i \text{ takes value } x_i).$$

- Function nodes to variables nodes:

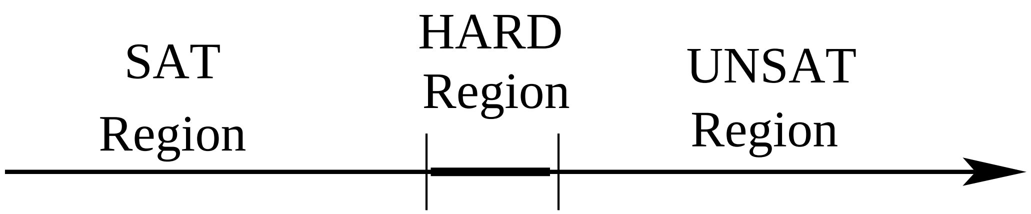
$$\mu_{a \to i}(x_i) = \mathrm{Pr}(\text{clause } a \text{ satisfied if } i = x_i).$$

- BP tries to solve fixed point equations for $\{\mu_{i \to a}, \mu_{a \to i}\}$.

Belief Propagation II

1. Where does Belief Propagation go wrong?

2. Local algorithm: BP has trouble with multiple clusters.



3. We want to choose a cluster: find frozen variables.

Survey Propagation I

1. Idea: Pass more complicated "global" messages.

2. BP passes probabilities that $i = x_i$.

3. Consider the survey over all clusters of the BP probabilities.

4. SP passes probability that BP probabilities will be 0,1 or something in between.

Survey Propagation II

1. We are quantizing the probability of a probability.

2. Thus we have three states:

   BP probability is 1
   BP probability is 0
   BP probability is between 0 and 1.

3. SP passes probabilities that variables take states $\{0, 1, *\}$.

4. SP is BP on an expanded state space.

Warning Propagation

1. Variable nodes to function nodes – a preference:

   $h_{i \to a} =$ Value node j takes to satisfy the most clauses other than a.

2. Function nodes to variable nodes – a warning:

   $u_{a \to i} = 1$ if no other variables in clause a satisfy a.

3. WP already quantized into three states:

   Variable i must be 1
   Variable i must be 0
   Variable i has no warnings: It is unfrozen.

Survey Propagation III

1. Idea: Consider the survey over all clusters, of WP messages.

2. This is already quantized: SP passes probabilities that

   Variable i must be 1

   Variable i must be 0

   Variable i can be $*$: it is unfrozen.

Survey Propagation: Discussion I

1. If SP converges we obtain a probability over clusters, that:

   Variable i must be 1

   Variable i must be 0

   Variable i can be $*$: it is unfrozen.

2. The Algorithm: Decimation

   (i) For each j, compute the probability that j is 1,-1.

   (ii) Find j that maximizes the difference in these two quantities.
   Freeze j.

   (iii) Repeat SP.

Survey Propagation: Discussion II

1. BP and WP both work well for trees, and in the easy SAT region.

2. On a tree, SP reduces to Warning Propagation.

3. SP seems to outperform BP and WP significantly in the HARD region, where α is close to α_c.

4. Empirical evidence suggests that $\alpha_c \approx 4.27$ and SP works "well" even near $\alpha \approx 4.252$.

Extensions: Backtracking

1. Backtracking: undo "frozen" variables.

2. Idea: On a tree, we can compute:

$$\mathcal{N} = \text{Number of satisfying assignments.}$$

   On a loopy graph, this provides an approximation.

3. At each step, fix the most "frozen" variable and unfreeze the variable that increases $\mathcal{N}$ the most.

Survey Propagation Equations I

1. Let $J_i^a = -1$ if variable i unnegated in a.

2. Define: $V(i), V_+(i), V_-(i), V(a), V_a^u(i)$.

3. WP recursive defining equations:

$$
\begin{aligned}
h_{j\to a} &= \left(\sum_{b\in V_+(j)\setminus a} u_{b\to j}\right) - \left(\sum_{b\in V_-(j)\setminus a} u_{b\to j}\right) \\
u_{a\to i} &= \prod_{j\in V(a)\setminus i} \theta(h_{j\to a} J_j^a),
\end{aligned}
$$

Survey Propagation Equations II

1. Instead of $u_{a\to i}$, we want the survey over the $\mathcal{N}$ clusters:

$$\begin{aligned} Q_{a\to i}(u) &= \frac{1}{\mathcal{N}} \sum \delta(u, u_{a\to i}) \\ &= (1-\eta_{a\to i})\delta(u,0) + \eta_{a\to i}\delta(u,1), \end{aligned}$$

2. Here we have set

$$\begin{aligned} \eta_{a\to i} &= \text{fraction of clusters where } u_{a\to i} \text{ equals 1} \\ &= \Pr(\text{for all } j \in V(a)\setminus i, \text{ we have } h_{j\to a} J_j(a) > 0) \\ &= \prod_{j\in V(a)\setminus i} \rho_{j\to a}. \end{aligned}$$

Survey Propagation Equations III

1. Let $Q(U)$ be the survey over all clusters, of all $u_{b\to j}$ for $j \in V(a) \setminus i$ and $b \in V(j) \setminus a$. Then,

$$\rho_{j\to a} = \sum_U Q(U)\, \theta\left(\sum_{b \in V_a^u(j) \setminus a} u_{b\to j} \right).$$

2. By independence of incoming messages, we have

$$\rho_{j\to a} = C_{j\to a} \sum_{\{u_{b\to j}\}} \left[\prod_{b\ \ inV(j)\setminus a} Q_{b\to j}(u_{b\to j}) \right] \theta\left(\sum_{b \in V_a^u(j)} u_{b\to j} \right) \prod_{b \in V_a^s(j)} \delta(u_{b\to j}, 0).$$

Survey Propagation Equations IV

1. We get the probabilities that variable j is fixed to 1. a value that does not satisfy a, 2. a value that does satisfy a, 3. is not fixed at all:

$$
\begin{aligned}
\Pi^u_{j\to a} &= \sum_{\{u_{b\to j}\}} \left[\prod_{b\in V(j)\setminus a} Q_{b\to j}(u_{b\to j})\right] \theta\left(\sum_{b\in V^u_a(j)} u_{b\to j}\right) \prod_{b\in V^s_a(j)} \delta(u_{b\to j},0) \\
\Pi^s_{j\to a} &= \sum_{\{u_{b\to j}\}} \left[\prod_{b\in V(j)\setminus a} Q_{b\to j}(u_{b\to j})\right] \theta\left(\sum_{b\in V^s_a(j)} u_{b\to j}\right) \prod_{b\in V^u_a(j)} \delta(u_{b\to j},0) \\
\Pi^0_{j\to a} &= \sum_{\{u_{b\to j}\}} \left[\prod_{b\in V(j)\setminus a} Q_{b\to j}(u_{b\to j})\right] \left[\prod_{b\in V(j)\setminus a} \delta(u_{b\to j},0)\right].
\end{aligned}
$$

Survey Propagation Equations V

1. Finally, our closed family of equations for our parameters $\eta_{a\to i}$, is:

$$
\begin{aligned}
\eta_{a\to i} &= \prod_{j\in V(a)\backslash a}\left[\frac{\Pi^u_{j\to a}}{\Pi^u_{j\to a}+\Pi^s_{j\to a}+\Pi^0_{j\to a}}\right] \\
\Pi^u_{j\to a} &= \left[1-\prod_{b\in V^u_a(j)}(1-\eta_{b\to j})\right]\prod_{b\in V^s_a(j)}(1-\eta_{b\to j}) \\
\Pi^s_{j\to a} &= \left[1-\prod_{b\in V^s_a(j)}(1-\eta_{b\to j})\right]\prod_{b\in V^u_a(j)}(1-\eta_{b\to j}) \\
\Pi^0_{j\to a} &= \prod_{b\in V(j)\backslash a}(1-\eta_{b\to j}).
\end{aligned}
$$

Some Questions

1. Where does the power of SP come from?

2. Can we view SP as a "tighter relaxation" than BP?

3. Can we understand the joker state in a polyhedral optimization framework?

4. What does SP buy us in decoding parity check codes?