SAT Factor Graphs

- CNF SAT formulas can be represented as bipartite graphs.
- Edges connect variable vertices to clause vertices.
- There are both negative and positive edges, represented as red and blue in our example graphs.

\[(\overline{X_1} \lor X_2 \lor \overline{X_3}) \land (X_1 \lor X_2 \lor X_3) \land (\overline{X_1} \lor \overline{X_2} \lor X_3)\]
When a variable takes on a specific value, outgoing edges which match that value are activated.

The formula is satisfied if all clauses can be reached from an activated edge.
Hard-SAT region

- A formula has $N$ variables and $M = \alpha N$ clauses.

- There is a phase shift in the range $3.9 < \alpha < 4.3$ causing satisfying assignments within the solution space to be separated by large Hamming distances.

- Searching the solution space becomes highly impractical!
Warning Propagation

Message passing algorithm:

- Clause vertex $a$ sends a boolean value $u_{a \rightarrow i}$ to the variable vertex $i$. If $u_{a \rightarrow i} = 1$, then $i$ should take on the value specified in clause $a$.

- Variable vertex $i$ sends a message to function node $a$ of the form:

$$h_{j \rightarrow a} = \left( \sum_{b \in V_p(j) \setminus a} u_{b \rightarrow j} \right) - \left( \sum_{b \in V_N(j) \setminus a} u_{b \rightarrow j} \right)$$

This is the number of warnings received from clauses in which $j$ is positive minus the number from clauses in which $j$ is negative.
Warning Propagation

- Attempts to find an approximation of the marginal probability for each variable that the variable will take on a specific value in a satisfying assignment.

- Guaranteed to provide a satisfying assignment when the factor graph is a tree.

- Generally fails to converge for high density formulas $\sim \alpha > 3.9$. 
Survey Propagation

- Message passing algorithm
- Three variable states are used: s, u, and 0
  - s corresponds to satisfying a clause
  - u corresponds to not satisfying the clause
  - 0 means that the variable is not yet constrained

- Partial or intermediate assignments are made up of values in \{0,1,*\}
- Variables that have been set to either 0 or 1 are constrained.
A message from variable $j$ to clause $a$ consists of three values

\[
\Pi_{j \rightarrow a}^u = \left[ 1 - \prod_{b \in V_{a}^{u}(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in V_{a}^{s}(j)} (1 - \eta_{b \rightarrow j}) \\
\Pi_{j \rightarrow a}^s = \left[ 1 - \prod_{b \in V_{a}^{s}(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in V_{a}^{u}(j)} (1 - \eta_{b \rightarrow j}) \\
\Pi_{j \rightarrow a}^0 = \prod_{b \in V(j) \setminus a} (1 - \eta_{b \rightarrow j})
\]

The product $\prod_{b \in V_{a}^{s}(j)} (1 - \eta_{b \rightarrow j})$ is the probability that no warning is sent to $j$ from clauses which would be satisfied by the value of $j$ specified in clause $a$. 
A message from clause $a$ to variable $i$ takes the form

$$\eta_{a \to i} = \prod_{j \in V(a) \setminus i} \left[ \frac{\prod_{j \to a}^u}{\prod_{j \to a}^u + \prod_{j \to a}^s + \prod_{j \to a}^0} \right]$$

This is the probability that a warning should be sent to $i$ from $a$.

The higher the value of $\eta_{a \to i}$, the more likely it is that $i$ must take on the value needed to satisfy clause $a$. 
Algorithm

**SP**
1. Randomly initialize every edge of the factor graph $\eta_{a \rightarrow i}$ in $[0,1]$
2. Do
   3. For each edge $(a,i)$ in a random permutation of the edges
   4. Compute $\Pi_{j \rightarrow a}$ for every $(a,j)$ edge and update $\eta_{a \rightarrow i}$ using the SP equations
   5. If the new values of $\eta$ have changed by a sufficiently small amount since the last iteration, return Converged
3. Return NotConverged

Marginalization-Decimation Approach: Run SP, evaluate biases, fix the variable with the largest bias, and repeat.
Implementation

- Graphviz (www.graphviz.org)
  - Graph visualization language *dot*
  - Programs DOT and NEATO apply heuristics to find good node configurations and create SVG files

- Survey Propagation code (Perl)
  - generates a random formula
  - creates a DOT factor graph
  - incrementally modifies it to reflect the state of the graph and the messages
Implementation

Nodes are marked with their bias, or the probability that they will take on the values \((-, +, \ast)\).

Edges are marked with their value for $\eta$, the probability the clause will send a warning to the attached variable.
Implementation
Implementation

- Simplifies graph as it iterates, removing edges which are not used to satisfy clauses
- This graph represents a solution to a 3-SAT instance on 4 variables and 14 clauses
Limitations

- Survey propagation often does not converge when $N$ is less than 1000.

- It is too hard to see a 1000 variable and 4000 clause factor graph, and the SVG (XML) files are ~5M.

Factor graph on $N = 1000$ and $\alpha = 4$