Chapter 2, part 2 relaxation

Chapter 2 covers topics to study the typical problems, data structures, and algorithms for inferring hierarchical and flat models.

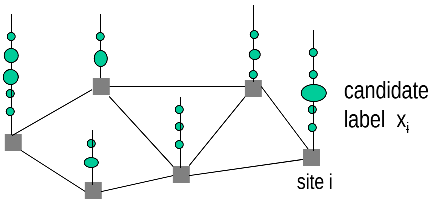
Part 1: Search on hierarchical Generative representations
Heuristic search algorithms on And-Or graphs
-– Best First Search, A*, Generalized Best First Search.

Part 2: Search on flat descriptive representations
2.1 Relaxation algorithm on line drawing interpretations
2.2 Belief propagation on poly-trees
2.3 Dynamic programming on chains.

Relaxation in a flat graph

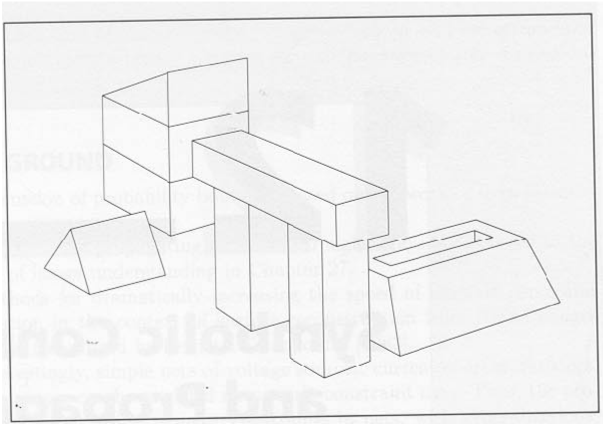
Common properties:
1. A graph representation G=<V, E>.
G could be directed, undirected, such as chain, tree, DAG, lattice, etc.

2. hard constraints or soft "energy" preference between adjacent vertices.



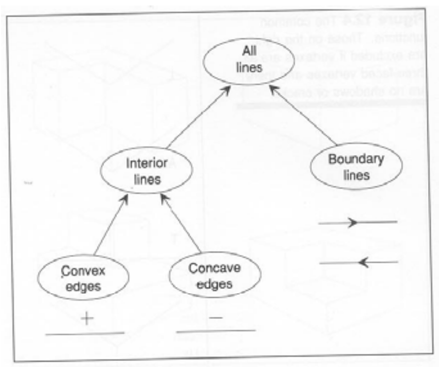
Examples: constraint-and-satisfaction, line drawings, graph partition/coloring, Turbo coding, image segmentation, scene labeling, …

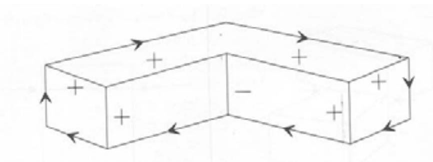
Ex 1: Symbolic interpretation of line drawings (Waltz, 1960s)



Line	Label
Convex	+
Concave	−
Boundary	>

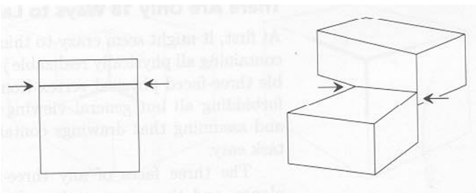
Labeling the lines with 4 symbols

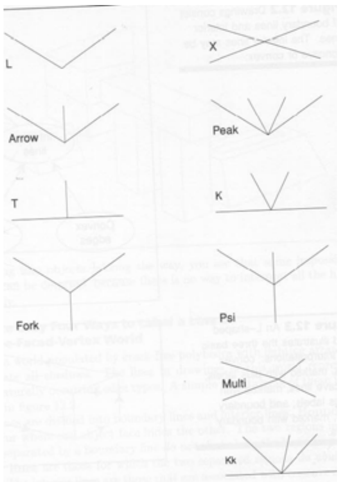




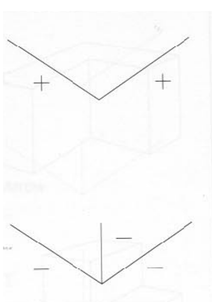
Excluding accidental views

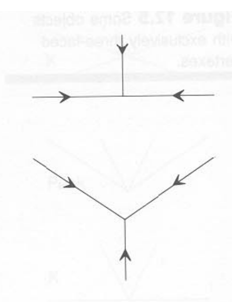
Accidental views cause alignments
(lines or junctions) which are unstable.

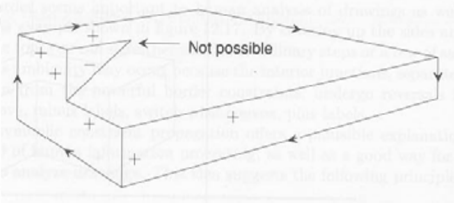




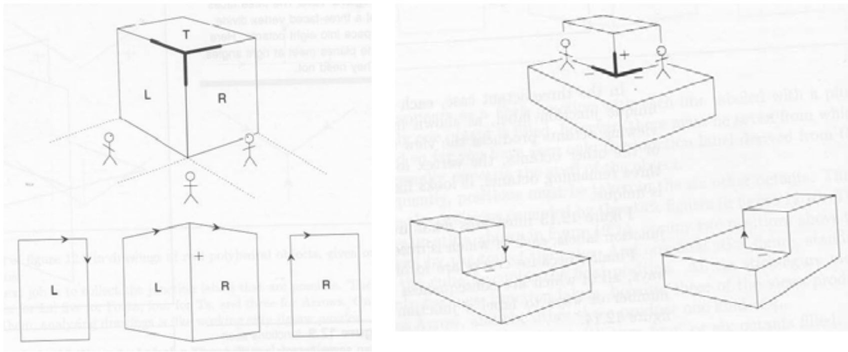
Examples of impossible realizations of corner by 3 faced object junctions





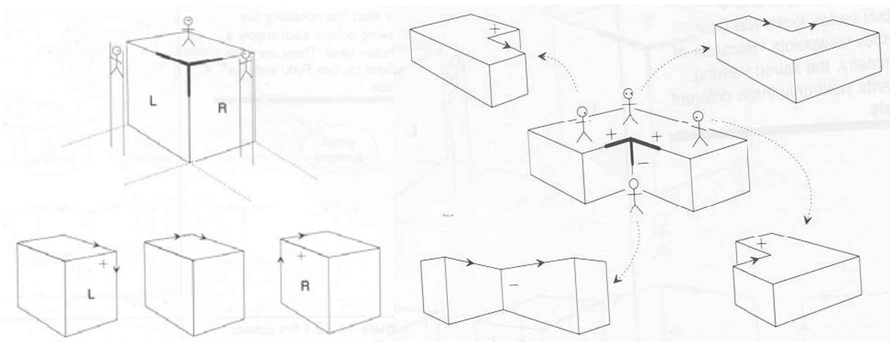


Finding all allowable junctions by considering all combinations of cubes in 8 quadrants in 3D



Finding all allowable junctions by considering all combinations of cubes in 8 quadrants in 3D

This is to exhaust all possible cases, all other non-accidental junctions will be equivalent to one of these junctions.



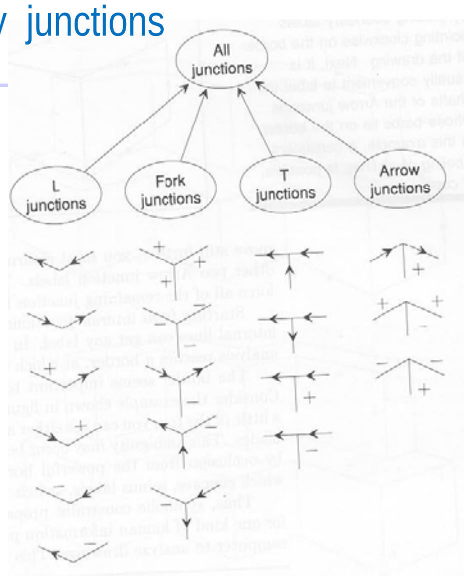
All allowable 2-way / 3-way junctions

18 valid junctions out of
208 possible combinations

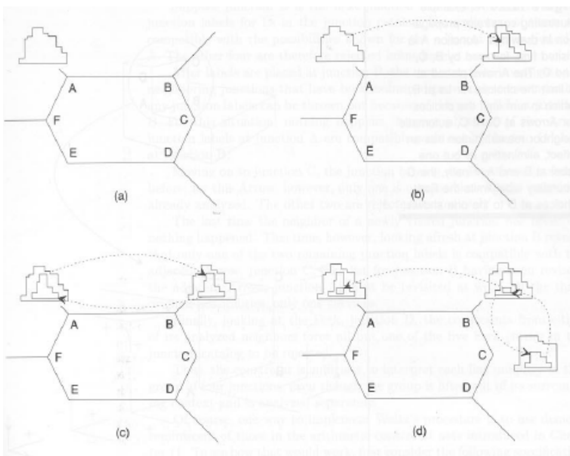
(208= 16 + 64 + 64 + 64)

Remarks: in general vision modeling, the space of parts/objects/scenes is combinatorial, but the true elements in real world is tiny. This leads to huge constraints as in line drawing interpretation.

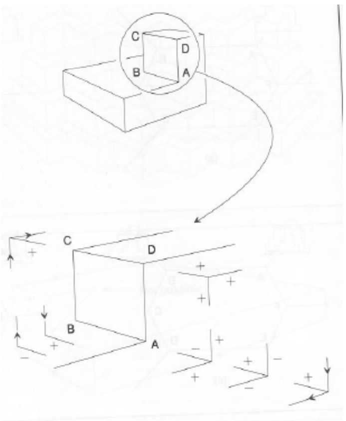
For an alternative way, one may manually label a number of representative block objects and then find the valid junctions from the label. This may not be exhaustive but is quite effective. In vision, we find vocabulary/elements this way As we couldn't possibly enumerate all cases.



Finding a valid interpretation through relaxation:
constraints-and-satisfaction



Finding a valid interpretation through relaxation:
constraints-and-satisfaction

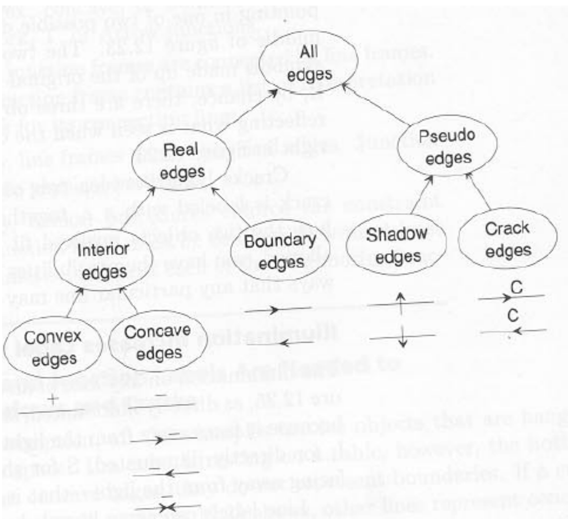


Note that what we compute here is still candidate labels at each vertex. When you have multiple ways to label an object, then we have to extract the solution.

Discussion:

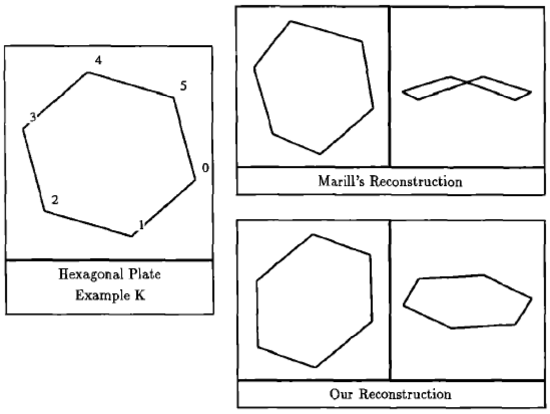
marginal beliefs vs. joint solutions

More complex labeling scheme by Waltz



Ex 2: 3D line drawing interpretation (Marill 91, Leclerc-Fishler'92)

1, Real 3D shapes
2, Soft-energy



Points	(0.96 -0.27) (0.24 -0.89) (-0.72 -0.61) (-0.96 0.27) (-0.24 0.89) (0.72 0.61)
Lines	(0 1) (1 2) (2 3) (3 4) (4 5) (5 0)
Faces	(0 1 2 3 4 5)

The energy function to optimize

$$E(\lambda) = \lambda SDA^2 + (1 - \lambda) DP$$

SDA stands for the standard deviation of all angles at a vertex.

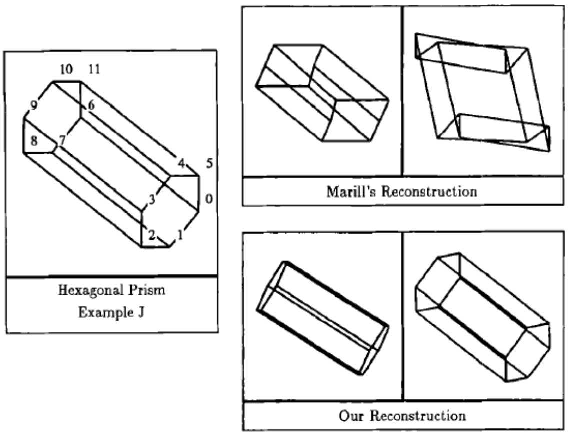
SDA enforces equal angels at each 3D vertex (thus to reach symmetry.

$$DP1 = \left((n - 2)\pi - \sum_j \alpha_j \right)^2$$

DP enforces co-planarity for angels in each face.

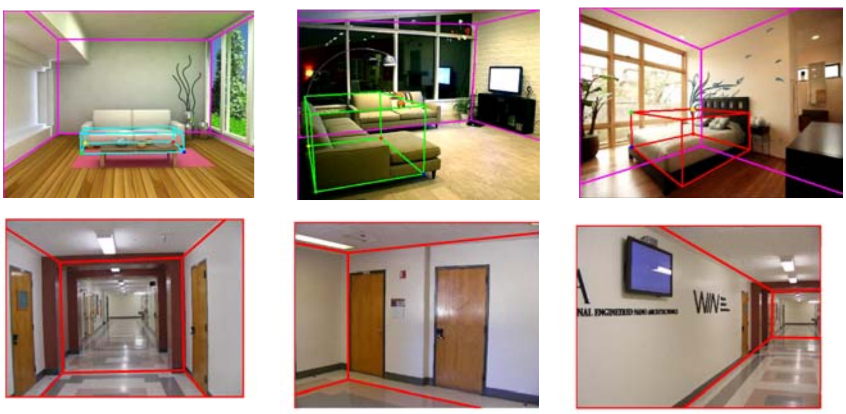
$$DP2 = \sum_j \left[1 - \left(\frac{(l_{j-1} \times l_j) \cdot (l_j \times l_{j+1})}{\|l_{j-1} \times l_j\| \; \|l_j \times l_{j+1}\|} \right) \right]^2$$

A typical result



Points	(1.97 -1.00) (1.32 -1.75) (0.67 -1.75) (0.68 -1.00) (1.34 -0.25) (1.98 -0.25) (-0.68 1.00) (-1.34 0.25) (-1.98 0.25) (-1.97 1.00) (-1.32 1.75) (-0.67 1.75)
Lines	(0 1) (1 2) (2 3) (3 4) (4 5) (5 0) (6 7) (7 8) (8 9) (9 10) (10 11) (11 6) (0 6) (1 7) (2 8) (3 9) (4 10) (5 11)
Faces	(0 1 7 6)(1 2 8 7)(2 3 9 8)(3 4 10 9)(4 5 11 10)(5 0 6 11)(0 1 2 3 4 5)(6 7 8 9 10 11)

More recent work on space reasoning on real images



Results from Yibiao Zhao, UCLA 2011.
other reference: Lee, Gupta, Hebert and Kanade, Estimating spatial layout of rooms using volumtric reasoning about objects and surfaces, NIPS 2010.

Ex 3: Belief propagation

In a Markov chain representation

$$b(x) = \alpha\, p(e^- \mid x)\, p(x \mid e^+) = \alpha\, \lambda(x)\, \pi(x)$$

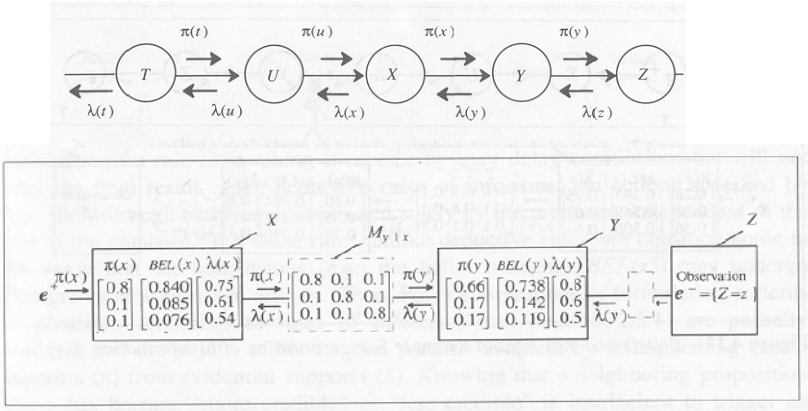


Figure 4.10. *Beliefs and messages in Example 5, after obtaining the laboratory report* λ(y).

BP is equivalent to DP in a chain

To get MAP solution (max product) from a belief network, e.g. 3-node graph, the BP algorithm is equivalent to the dynamic programming.

$$\hat{x}_1 = \arg\max_{x_1} \phi(x_1) \max_{x_2}\left[\phi(x_2)\psi(x_1,x_2) \max_{x_3}\left(\phi(x_3)\psi(x_2,x_3)\right)\right]$$

$$\hat{x}_2 = \arg\max_{x_2} \phi(x_2)\left[\max_{x_1}\left(\phi(x_1)\psi(x_1,x_2)\right)\right]\left[\max_{x_3}\left(\phi(x_3)\psi(x_2,x_3)\right)\right]$$

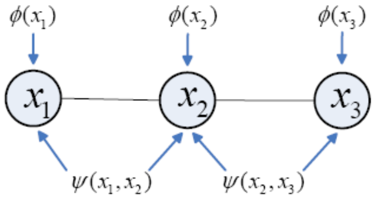
$$\hat{x}_3 = \arg\max_{x_3} \phi(x_3) \max_{x_2}\left[\phi(x_2)\psi(x_2,x_3) \max_{x_1}\left(\phi(x_1)\psi(x_1,x_2)\right)\right]$$

on a flat graph (no-loop)

For pairwise MRF's, the joint prob. distribution for {x} can be factorized

$$p(\{x\}) = \frac{1}{Z}\prod_{i,j}\psi_{ij}(x_i, x_j)\prod_i \phi_i(x_i)$$

where $\psi_{ij}(x_i, x_j)$ tells internal bound between node i and j, and $\phi_i(x_i)$ indicates external evidence at node i.



Message passing

Belief is the marginal posterior probability of a node x over all evidence e

b(x) = prob(x | e) = α p(e^- | x) p(x | e^+) = α λ(x) π(x)

Messages 'm' are introduced to pass information between nodes in BP network. The belief (marginal posterior) at a node i is computed as follows:

$$b_i(x_i) = \alpha\phi_i(x_i)\prod_{j\in N(i)} m_{ji}(x_i) \tag{5}$$

and the joint belief (joint marginal posterior) of a pair of neighboring nodes i and j is:

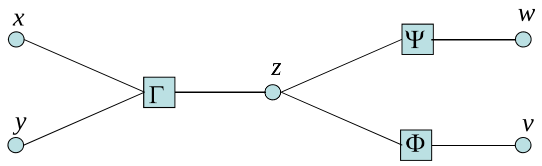
$$b_{ij}(x_i, x_j) = \beta\psi_{ij}(x_i, x_j)\phi_i(x_i)\phi_j(x_j)\prod_{k\in N(i)} m_{ki}(x_i)\prod_{l\in N(j)} m_{lj}(x_j) \tag{6}$$

the message from nodes j to i is:

$$m_{ji}(x_i) \leftarrow \sum_{x_i}\phi_i(x_i)\psi_{ij}(x_i, x_j)\prod_{k\in N(i)\setminus j} m_{ki}(x_i)$$

BP on Factor Graphs

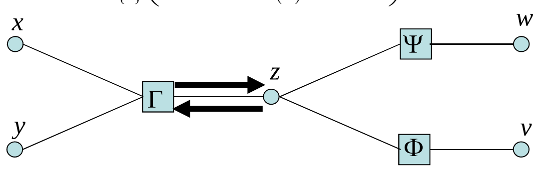
$$f\left(x,y,z,w,v\right)=\Gamma\left(x,y,z\right)\Psi\left(z,w\right)\Phi\left(z,v\right)$$



Belief Propagation Algorithm

- The messages sent across the edges are the "local" marginal pdfs.
- The marginal pdf of *z* is the multiplication of all incoming messages

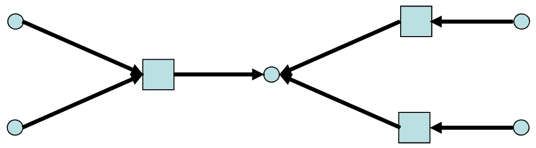
$$m_{\Gamma\to z}=\sum_{\sim\{z\}}\left(\Gamma\left(X\right)\prod_{a\in N(\Gamma)\backslash z} m_{a\to\Gamma}\right)$$



$$m_{z\to\Gamma}=\prod_{f\in N(z)\backslash\Gamma} m_{f\to z}$$

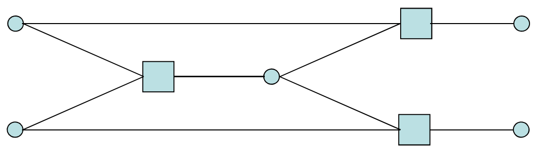
Belief Propagation

- Nodes generate messages and send them to their neighbors
- BP provides the right answer if the graph has no loops
- Message propagation schedule is well defined



Loopy BP

- If graph has loops:
  – BP is no longer guaranteed to converge
  – If it converges it is not necessarily to the right answer
  – BP becomes an iterative algorithm where the message passing schedule is not well defined
- However, loopy BP has been shown to perform very well for many applications



Visiting order of nodes in the flat graph

The visiting order (scheduling) is an interesting problem.

1, Discuss the case of global constraints and random belief updates

2, The visiting order in an And-Or graph can also be formulated in a BP algorithm.

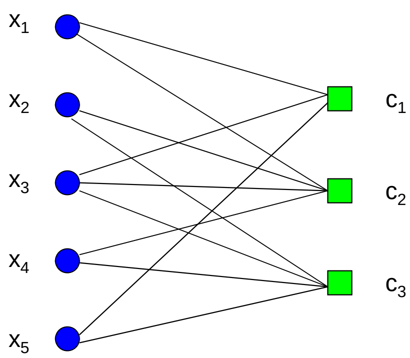
Example: applying BP to channel coding

(example from Dr. Andres I. Vila Casado EE, UCLA)

A binary-channel code can be seen as a constraint-and –satisfaction problem

$$g(x_1, x_2, \dots x_n) \in \{\ X:\ x_1 \otimes x_3 \otimes x_5 = 0; x_1 \otimes x_2 \otimes x_3 \otimes x_4 = 0\ \}$$

	x_1	x_2	x_3	x_4	x_5
c_1	1	0	1	0	1
c_2	1	1	1	1	0
c_3	0	1	1	1	1



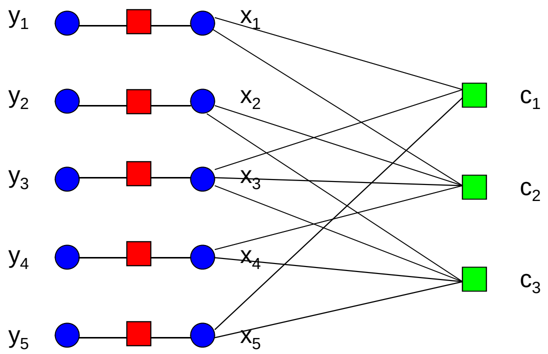
Each row is a constraint

Loopy BP for Channel decoding

Global function is the joint-probability of the received signal y and the true codeword x

$$p(x_1,\dots,x_n,y_1,\dots,y_n) = p(x_1,\dots,x_n)\cdot p(y_1,\dots,y_n|x_1,\dots,x_n)$$

$$p(y_1,\dots,y_n|x_1,\dots,x_n)=\prod_{i=1}^{n} p(y_i|x_i) \Rightarrow p(x_1,\dots,x_n,y_1,\dots,y_n)=\gamma[(x_1,\dots,x_n)\in C]\cdot\prod_{i=1}^{n} p(y_i|x_i)$$



(Vila Casado)

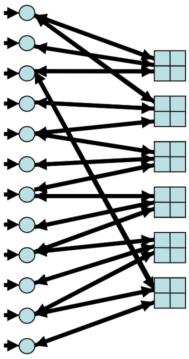
Message-passing scheduling problem

- How to schedule the message-passing algorithm in decoding?
- Pre-determined scheduling:
  - Flooding
  - Standard Sequential Scheduling
- Informed Dynamic Scheduling (IDS):
  - Use the current state of messages in the graph to find the next message to be propagated

(Vila Casado)

Simultaneous (Flooding) Schedule

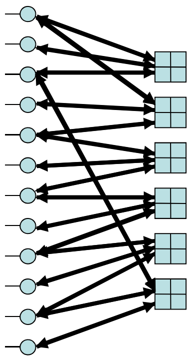
On every iteration
- All variable nodes are simultaneously updated
- All check nodes are simultaneously updated



(Vila Casado)

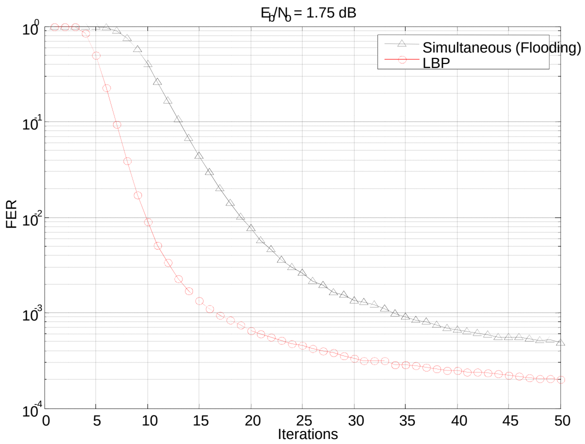
Standard Sequential Schedule (SSS)

Update messages sequentially:



(Vila Casado)

Speed comparison



Andres I. Vila Casado, UCLA

Historical Milestones

Line drawing labeling (Waltz, 1972)
Relaxation-Labeling algorithm (Rosenfeld, Hummel and Zucker, 1976)
Belief propagation (Pearl, 1982)
Gibbs sampler (Geman and Geman, 1984)
Cluster sampling (Swendsen-Wang,1987)

For special cases:
On Chains:
Dynamic programming (Bellman 1957)
= Belief propagation = Exact sampling (Gibbs sampler)