Tree-reweighted max-product and LP relaxation: Algorithmic connections and probabilistic analysis

Martin Wainwright
Department of Electrical Engineering and Computer Science
Department of Statistics
UC Berkeley, CA
Email: wainwrig@{eecs,stat}.berkeley.edu

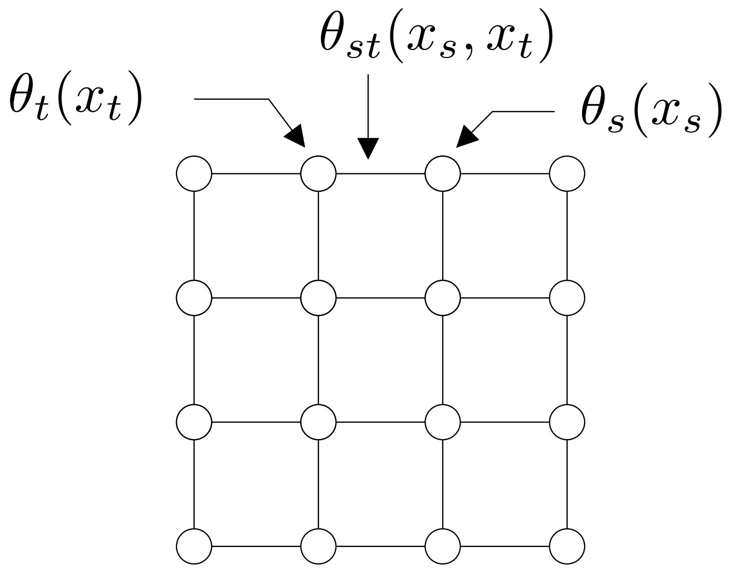
Based on joint papers with:

Tommi Jaakkola, Alan Willsky (MIT)
Vladimir Kolmogorov (UCL)
Costis Daskalakis, Alex Dimakis, Richard Karp (UC Berkeley)

Introduction

- **message-passing:** now standard method in various fields (coding, physics, computer vision, learning, computational biology....)
- **linear programming (LP) relaxation:** standard method in theoretical computer science, operations research, math programming etc.
- **fruitful connections** between these two frameworks
- some useful features of LP relaxation:
  - certificates of correctness
  - hierarchies of relaxations (guaranteed improvement; increased cost)
- some useful features of message-passing:
  - cheap, scalable algorithms; distributed in nature
  - easy to implement (both in software and hardware)
  - finite convergence for LP solving

MAP optimization in undirected graphical models



- undirected graph $G = (V, E)$
- $X_s \equiv$ random variable at node s taking values $x_s \in \mathcal{X}_s$
- $\theta_s(x_s) \equiv$ observation term
- $\theta_{st}(x_s, x_t) \equiv$ coupling term

- overall distribution decomposes additively on graph cliques:

$$p(x; \theta) \quad \propto \quad \Big\{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \Big\}$$

- mode or maximum a posteriori (MAP) estimate:

$$x^* \quad \in \arg\max_{x \in \mathcal{X}^n} \Big\{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \Big\}.$$

Outline

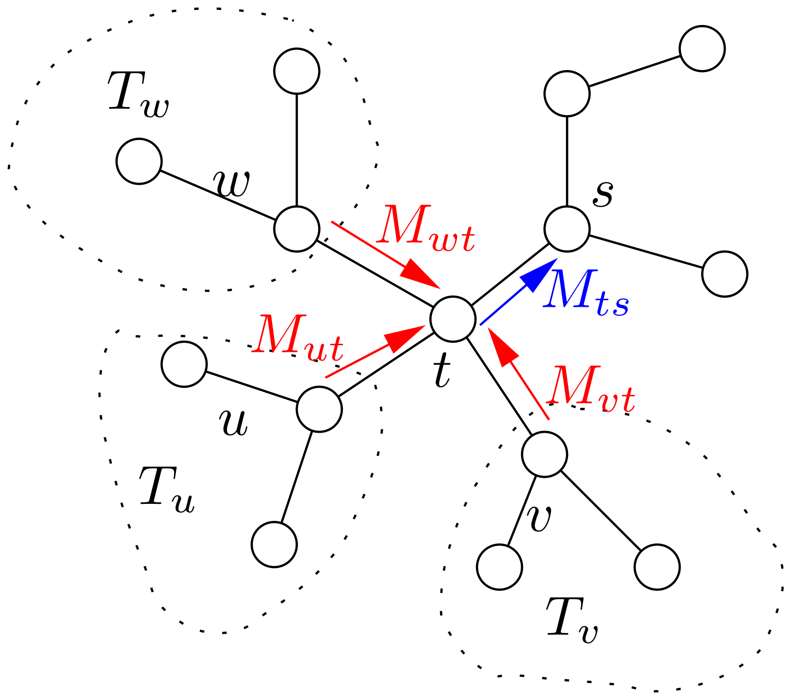
1. From ordinary to reweighted max-product
   (a) Deficiencies of ordinary max-product
   (b) Reweighted max-product
   (c) Connection to LP relaxation

2. Probabilistic analysis of LP relaxation
   (a) Motivation: worst-case versus average-case analysis
   (b) Graphical models and LP relaxations for decoding
   (c) Probabilistic guarantees on performance

3. Open directions/questions

Standard message-passing algorithms: On trees

Exact for trees, but approximate for graphs with cycles.



$M_{ts} \quad \equiv \quad$ message from node t to s

$\mathcal{N}(t) \quad \equiv \quad$ neighbors of node t

Sum-product: for marginals

(generalizes $\alpha - \beta$ algorithm; Kalman filter)

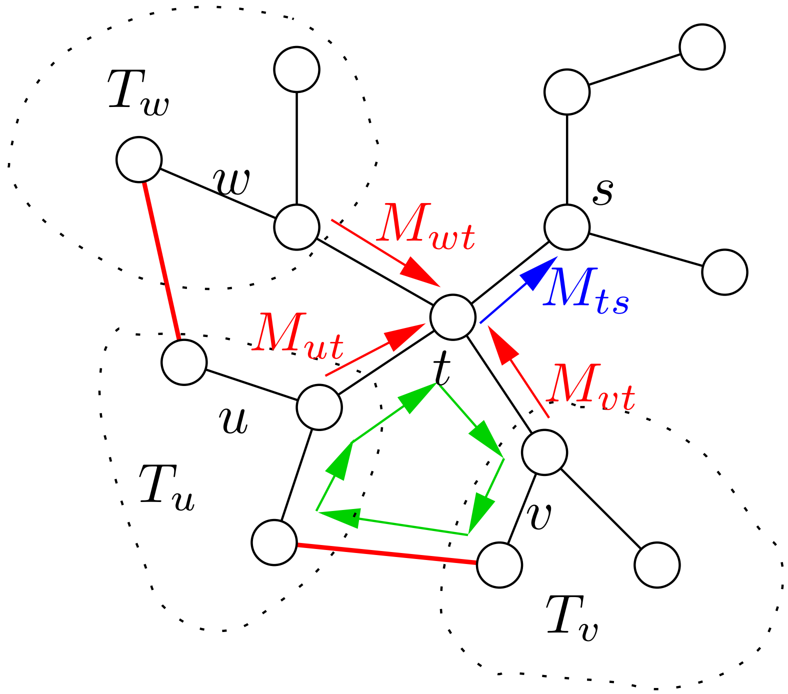
Max-product: for MAP configurations

(generalizes Viterbi algorithm)

Update: $\quad \mathbf{M_{ts}(x_s)} \leftarrow \max_{x_t' \in \mathcal{X}_t} \left\{ \exp \left[\theta_{st}(x_s, x_t') + \theta_t(x_t') \right] \prod_{v \in \mathcal{N}(t) \backslash s} \mathbf{M_{vt}(x_t)} \right\}$.

Standard message-passing algorithms: With cycles

Exact for trees, but approximate for graphs with cycles.



M_{ts}	$\equiv$	message from node t to s
$\mathcal{N}(t)$	$\equiv$	neighbors of node t

Sum-product: for marginals

Max-product: for modes

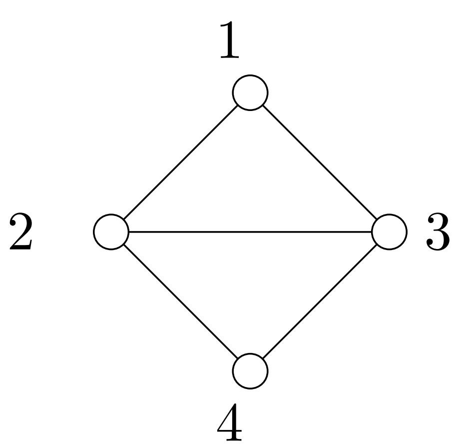
Update: $\mathbf{M_{ts}(x_s)} \leftarrow \max_{x'_t \in \mathcal{X}_t} \Big\{ \exp\Big[\theta_{st}(x_s, x'_t) \,+\, \theta_t(x'_t)\Big] \prod_{v \in \mathcal{N}(t) \backslash s} \mathbf{M_{vt}(x_t)} \Big\}$.

Some previous theory on ordinary max-product

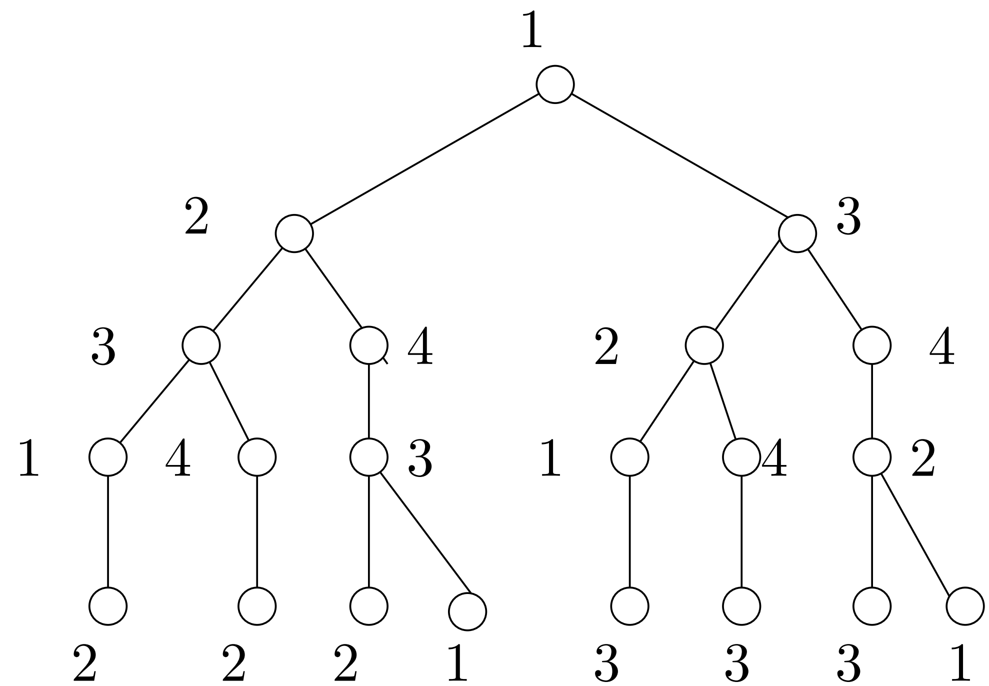
- well-known to be optimal on trees
- analysis of graphs with large girth (Gallager, 1963; many others, 1990s onwards)
- single-cycle graphs (Aji & McEliece, 1998; Horn, 1999; Weiss, 1998)
- local optimality guarantees:
  - "tree-plus-loop" neighborhoods (Weiss & Freeman, 2001)
  - strengthened optimality results and computable error bounds (Wainwright et al., 2003)
- max. weight bipartite matching (Bayati, Shah & Sharma, 2005)

Standard analysis via computation tree

- standard tool: computation tree of message-passing updates
  (Gallager, 1963; Weiss, 2001; Richardson & Urbanke, 2001)



(a) Original graph



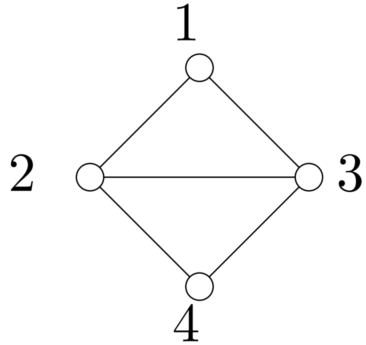
(b) Computation tree (4 iterations)

- level t of tree: all nodes whose messages reach the root (node 1) after t iterations of message-passing

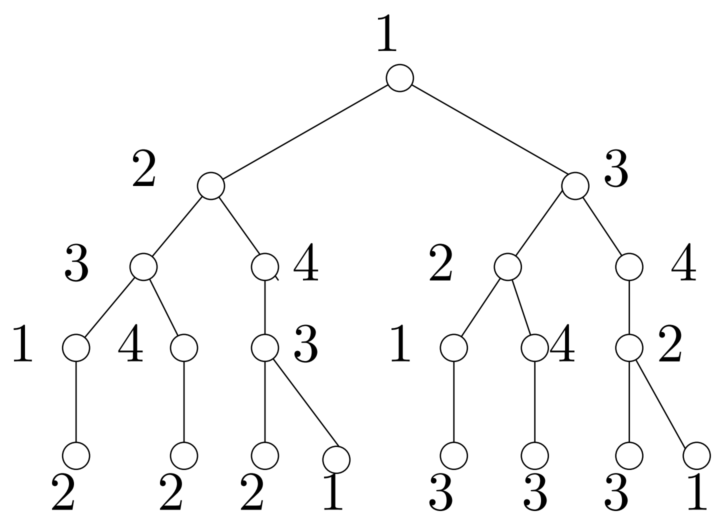
Illustration: Non-exactness of standard max-product

Intuition:

- max-product solves (exactly) modified problem on computation tree
- edge/nodes *not equally weighted* ⇒ incorrectness of max-product



(a) Diamond graph G_{dia}



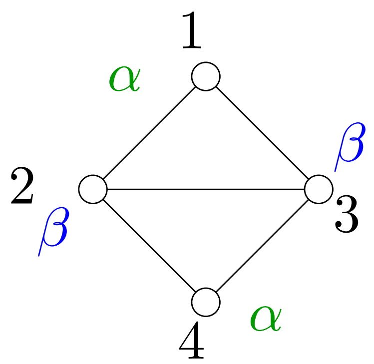
(b) Computation tree (4 iterations)

- for example: asymptotic node fractions in this computation tree:

$$\begin{bmatrix} f(1) & f(2) & f(3) & f(4) \end{bmatrix} = \begin{bmatrix} 0.2393 & 0.2607 & 0.2607 & 0.2393 \end{bmatrix}$$

A whole family of non-exact examples

- consider the following integer program on G_{dia}:



$$\theta_s(x_s) \quad \begin{cases} \alpha x_s & \text{if } s = 1 \text{ or } s = 4 \\ \beta x_s & \text{if } s = 2 \text{ or } s = 3 \end{cases}$$

$$\theta_{st}(x_s, x_t) \quad = \quad \begin{cases} -\gamma & \text{if } x_s \neq x_t \\ 0 & \text{otherwise} \end{cases}$$

- for γ sufficiently large, optimal solution is always either $0^4 = [0\,0\,0\,0]$ or $1^4 = [1\,1\,1\,1]$.

- max-product and optimum give *different* decision boundaries:

Optimum boundary: $\quad \widehat{\mathbf{x}} = \begin{cases} 1^4 & \text{if } 0.25\alpha + 0.25\beta \geq 0 \\ 0^4 & \text{otherwise} \end{cases}$

Max-product boundary: $\quad \widehat{\mathbf{x}} = \begin{cases} 1^4 & \text{if } 0.2393\alpha + 0.2607\beta \geq 0 \\ 0^4 & \text{otherwise} \end{cases}$

Tree-reweighted max-product algorithm

(Wainwright, Jaakkola & Willsky, 2002)

Message update from node t to node s:

$$M_{ts}(x_s) \quad \leftarrow \quad \kappa \max_{x'_t \in \mathcal{X}_t} \left\{ \exp\Big[\underbrace{\frac{\theta_{st}(x_s, x'_t)}{\rho_{st}}}_{\text{reweighted edge}} + \theta_t(x'_t) \Big] \frac{\prod_{v \in \mathcal{N}(t) \backslash s} \overbrace{\big[M_{vt}(x_t)\big]^{\rho_{vt}}}^{\text{reweighted messages}}}{\underbrace{\big[M_{st}(x_t)\big]^{(1-\rho_{ts})}}_{\text{opposite message}}} \right\}.$$

Properties:

1. Modified updates remain *distributed* and *purely local* over the graph.
2. Key differences:
   - Messages are reweighted with $\rho_{st} \in [0,1]$.
   - Potential on edge (s,t) is rescaled by $\rho_{st} \in [0,1]$.
   - Update involves the reverse direction edge.
3. The choice $\rho_{st} = 1$ for all edges (s,t) recovers standard update.

TRW max-product never "lies"

Set-up: A fixed point ν^* satisfies strong tree agreement (STA) if there exists a configuration $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ such that

$$\underbrace{x_s^* \in \arg\max_{x_s} \nu_s^*(x_s),}_{\text{Node optimality}} \qquad \underbrace{(x_s^*, x_t^*) \in \arg\max_{x_s, x_t} \nu_{st}^*(x_s, x_t)}_{\text{Edge-wise optimality}}$$

Theorem 1: For "suitable" edge weights ρ_{st}: (Wainwright et al., 2003):

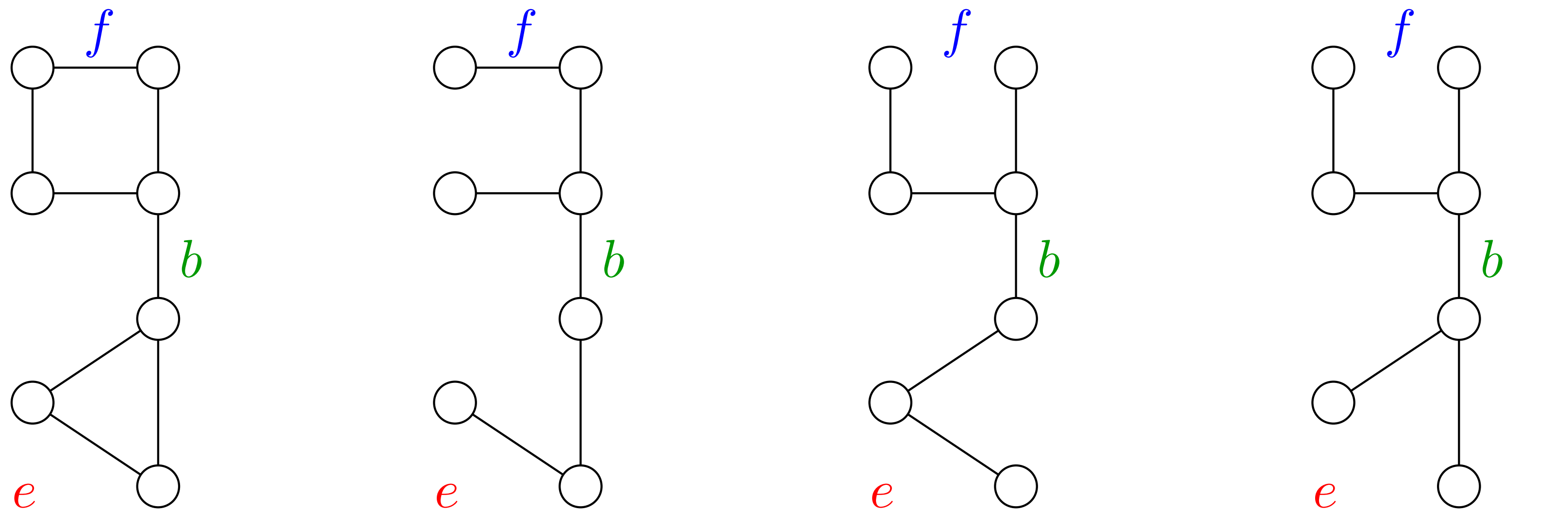
(a) Any STA configuration $\mathbf{x}^*$ is provably MAP-optimal for the graph with cycles.

(b) Any STA fixed point is a dual-optimal solution to a certain "tree-based" linear programming relaxation.

Hence, TRW max-product acknowledges failure by *lack of strong tree agreement.*

Edge appearance probabilities

Experiment: What is the probability ρ_e that a given edge $e \in E$ belongs to a tree T drawn randomly under $\boldsymbol{\rho}$?



(a) Original (b) $\rho(T^1) = \frac{1}{3}$ (c) $\rho(T^2) = \frac{1}{3}$ (d) $\rho(T^3) = \frac{1}{3}$

In this example: $\rho_b = 1$; $\rho_e = \frac{2}{3}$; $\rho_f = \frac{1}{3}$.

The vector $\boldsymbol{\rho_e} = \{ \rho_e \mid e \in E \}$ must belong to the *spanning tree polytope*, denoted $\mathbb{T}(G)$. (Edmonds, 1971)

Basic idea: convex combinations of trees

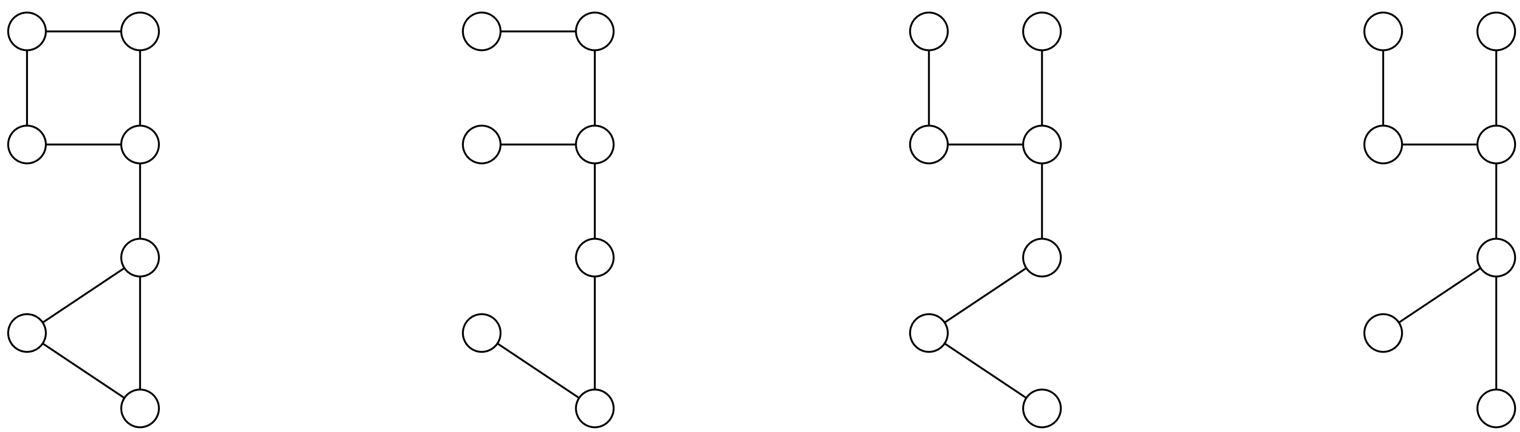
Observation: Easy to find its MAP-optimal configurations on trees:

$$\mathrm{OPT}(\theta(T)) \quad := \quad \big\{\mathbf{x} \in \mathcal{X}^n \mid \mathbf{x} \text{ is MAP-optimal for } p(\mathbf{x};\theta(T))\big\}.$$

Idea: Approximate original problem by a convex combination of trees.

$\boldsymbol{\rho} = \{\rho(T)\}$ $\equiv$ probability distribution over spanning trees

$\theta(T)$ $\equiv$ tree-structured parameter vector



$$* \qquad \theta^* \quad = \quad \rho(T^1)\theta(T^1) \quad + \quad \rho(T^2)\theta(T^2) \quad + \quad \rho(T^3)\theta(T^3)$$

$$\dagger \qquad \mathrm{OPT}(\theta^*) \quad \supseteq \quad \mathrm{OPT}(\theta(T^1)) \quad \cap \quad \mathrm{OPT}(\theta(T^2)) \quad \cap \quad \mathrm{OPT}(\theta(T^3)).$$

Dual perspective: linear programming relaxation

- Upper bound maintained by reweighted message-passing:

$$\max_{\mathbf{x} \in \mathcal{X}^N} \langle \theta^*, \phi(\mathbf{x}) \rangle \quad \leq \quad \sum_{T \in \mathfrak{T}} \rho(T) \max_{\mathbf{x} \in \mathcal{X}^N} \langle \theta(T), \phi(\mathbf{x}) \rangle$$

- Dual of finding optimal upper bound $\equiv$ tree-based LP relaxation:

$$\max_{\mathbf{x} \in \mathcal{X}^N} \langle \theta^*, \phi(\mathbf{x}) \rangle \quad \leq \quad \max_{\mu \in \mathrm{LOCAL}(G)} \langle \mu, \phi(\mathbf{x}) \rangle$$

- TRW-MP algorithm fixed points specify LP optimum:
  - whenever strong tree agreement holds (WaiJaaWil05)
  - for any binary problem (KolWai05)
  - ....but TRW-MP not solving LP in general! (Kol05)

Various connections and extensions

- edge-based updates and max-sum diffusion (Schlesinger et al., 1960s)
- binary QPs: roof duality equivalent to relaxation using LOCAL(G) (Hammer et al., 1984; Boros et al., 1990)
- natural hierarchy of LP relaxations based on treewidth:

$$\text{MARG}(G) = \text{LOCAL}_t(G) \subset \text{LOCAL}_{t-1}(G) \subset \ldots \subset \text{LOCAL}_1(G)$$

- treewidth hierarchy: equivalent to Boros et al. (1990) and Sherali-Adams (1990) hierarchies for binary problems
- other approaches with links to first-order LOCAL(G) LP relaxation:
  - sequential TRW and conv. guarantees (Kolmogorov, 2005)
  - convex free energies (Weiss et al., 2007)
  - sub-gradients (Feldman et al, 2003; Komodakis et al., 2007)
  - proximal projections (Ravikumar et al., 2008)

§2. Probabilistic analysis of LP relaxation

Classical complexity theory:

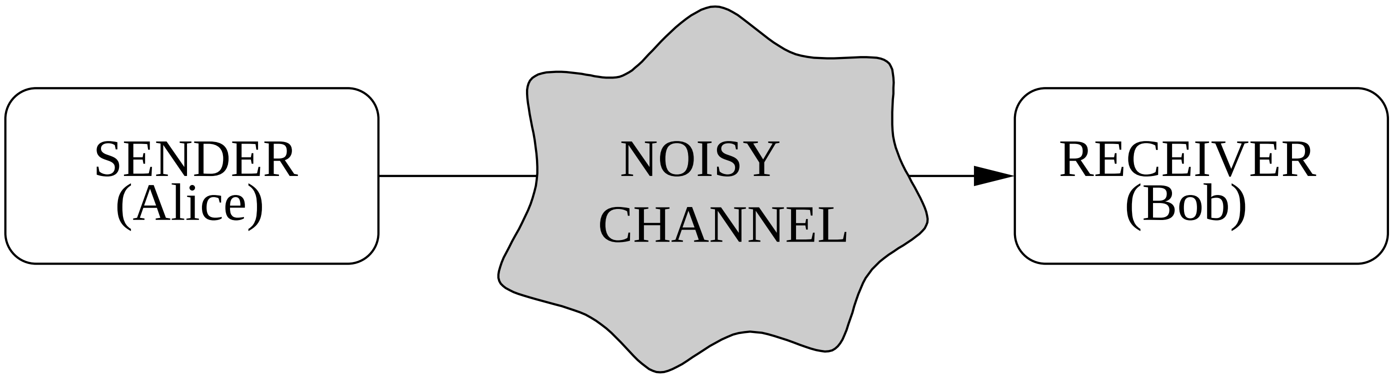
- worst-case or adversarial in nature
- problem class is “hard” if there *exists some instance* that is difficult to solve
- concern: how relevant are hard instances to practical applications?

Average-case analysis:

- consider random ensembles of instances
- natural ensembles in many application domains: statistical physics, communication, signal processing, vision, machine learning
- **Goal:** show that a method succeeds with high probability for a randomly chosen instance

Motivation: Reliable communication under noisy conditions

- consider two "people" (Alice and Bob) who would like to communicate (i.e., transmit information)
- channel: any mechanism by which Alice and Bob can communicate



Fundamental question: How can Alice transmit information *reliably* to Bob over an *unreliable* channel?

Wide range of applications: satellite communication; wireless networks; product barcodes; computer hard drives; neural communication

Error-control: Binary linear codes

- information represented by bit strings $x \in \{0,1\}^n$
- Alice introduces redundancy into transmission by sending *only* a subset $\mathbb{C}$ of all possible 2^n binary strings
  - **Example:** parity checks: require subsets of bits to be even parity

$$x_1 \oplus x_7 \oplus x_8 \quad = \quad 0.$$

- a binary linear code $\mathbb{C}$ is the null space of parity check matrix

$$\mathbb{C} \quad := \quad \{x \in \{0,1\}^n \;\mid\; Hx = 0\}$$

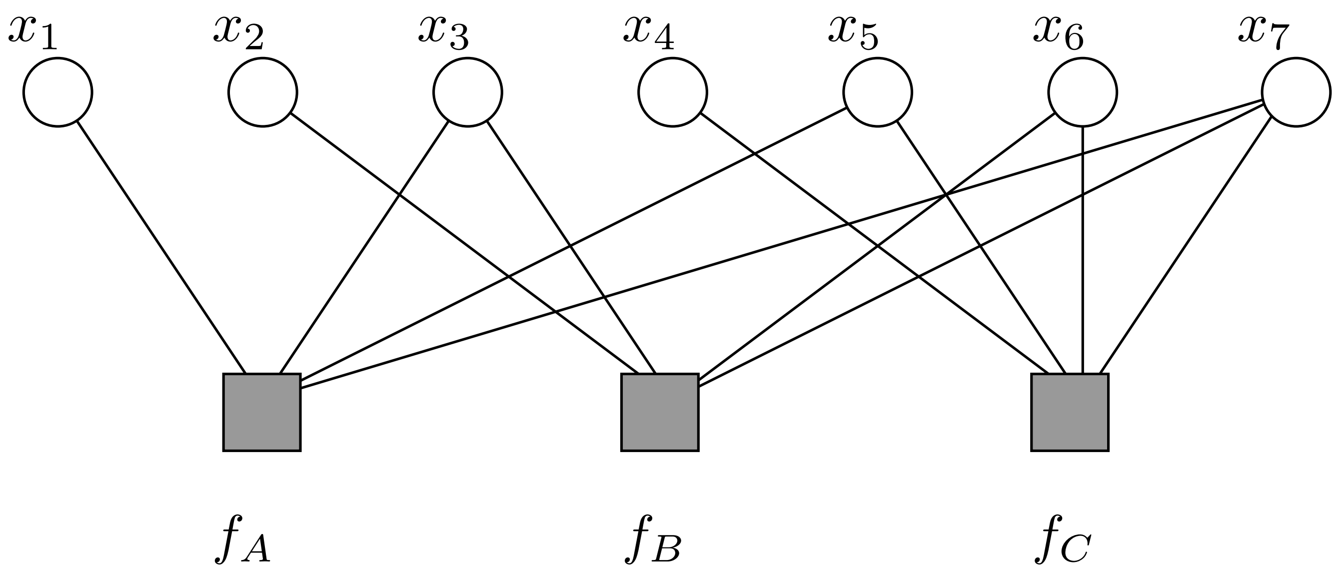
  where $H \in \{0,1\}^{m \times n}$ is the *parity check matrix*

- information rate $R = 1 - \frac{m}{n}$, since parity check matrix reduces degrees of freedom by m

(Shannon, 1940s)

Factor graph representation

Example: Parity check matrix: $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$



- square nodes ■ represent parity checks (rows of H)
- circular nodes ○ represent code bits (columns of H)

Error-control decoding

- observe a corrupted version of each transmitted bit:

$$y_i \quad = \quad \begin{cases} x_i & \text{with probability } 1-p \\ 1-x_i & \text{with probability } p \end{cases}$$

- optimal decoding corresponds to finding the nearest codeword

$$x^* \; = \; \arg \min_{x \in \{0,1\}^n} \|y - x\|_1 \qquad \text{H x} = 0$$

- can be formulated as an (intractable) LP over codeword polytope:

$$x^* \quad = \quad \arg \min_{\mu \in \mathrm{CH}(\mathbb{C})} \sum_{i=1}^{n} \gamma_i \mu_i$$

where

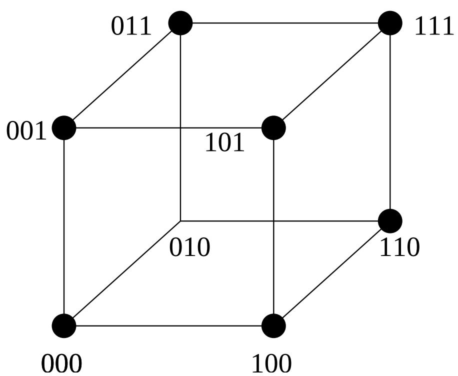
$$\gamma_i \quad = \quad \begin{cases} \log \frac{p}{1-p} & \text{if } y_i = 1 \\ -\log \frac{p}{1-p} & \text{if } y_i = 0 \end{cases}$$

- optimal decoding NP-complete in general (Berlekamp et al., 1978)

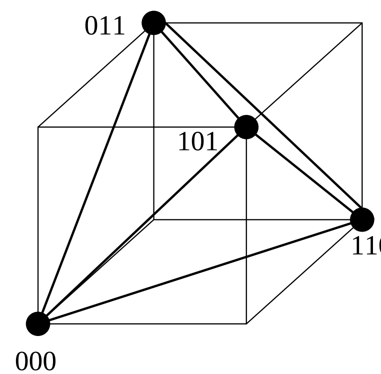
Codeword polytope

Definition: The *codeword polytope* $\mathrm{CH}(\mathbb{C}) \subseteq [0,1]^n$ is the convex hull of all codewords

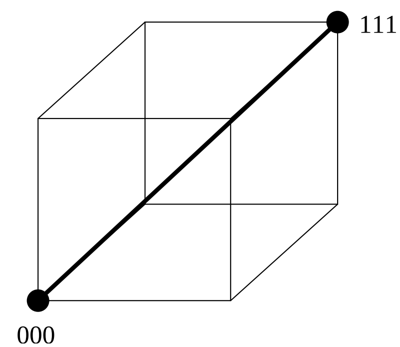
$$\mathrm{CH}(\mathbb{C}) = \left\{ \mu \in [0,1]^n \mid \mu_s = \sum_{\mathbf{x} \in \mathbb{C}} p(\mathbf{x})\, x_s \right\}$$



(a) Uncoded



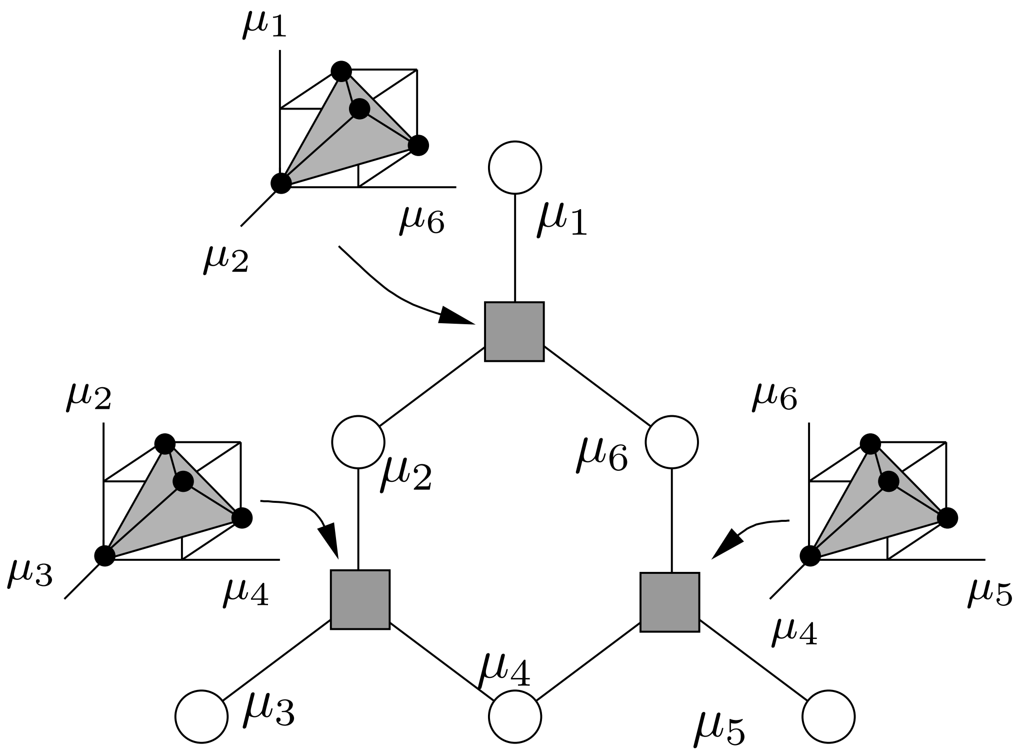
(b) One check



(c) Two checks

- the codeword polytope is always contained within the unit hypercube $[0,1]^n$
- vertices correspond to codewords

First-order relaxation for decoding



- each parity check $a \in C$ defines a *local codeword polytope* $\text{LOCAL}_1(a)$
- first-order relaxation obtained by imposing all local constraints:

$$\text{LOCAL}_1(\mathbb{C}) \;:=\; \cap_{a \in C} \, \text{LOCAL}_1(a).$$

(Feldman, Wainwright & Karger, 2003)

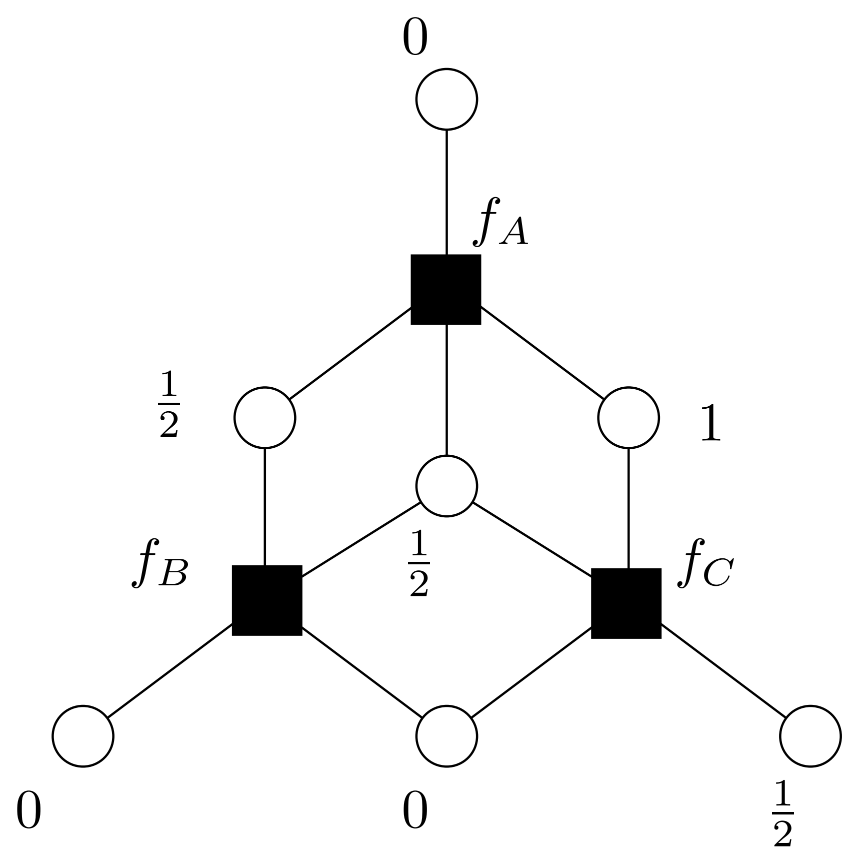
Illustration of fractional vertex (pseudocodeword)

Check A:

$$\begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Check B:

$$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

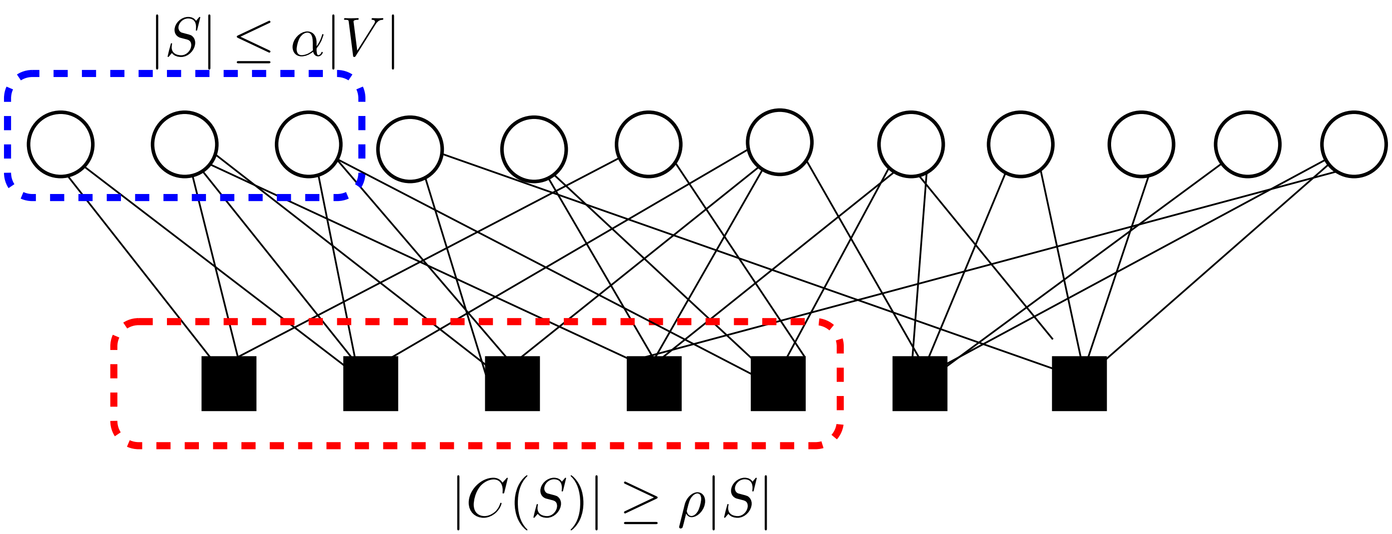


The pseudocodeword is locally-consistent for each check $\Longrightarrow$ it belongs to the first-order relaxed polytope $\mathrm{LOCAL}_1(\mathbb{C})$.

Codes based on expander graphs

- previous work on expander codes (e.g., SipSpi02; BurMil02; BarZem02)

- graph expansion: yields stronger results beyond girth-based analysis



- **Definition:** Let $\alpha \in (0, 1)$. A factor graph $G = (V, C, E)$ is a (α, ρ)-*expander* if for all subsets $S \subset V$ with $|S| \leq \alpha|V|$, at least $\rho|S|$ check nodes are incident to S

Worst-case constant fraction for expanders

Theorem: Let $\mathbb{C}$ be an LDPC described by a factor graph G with regular variable (bit) degree d_v. Suppose that G is an $(\alpha, \delta d_v)$-expander, where $\delta > 2/3 + 1/(3d_v)$ and δd_v is an integer.

Then the LP decoder can correct any pattern of $\frac{3\delta - 2}{2\delta - 1}(\alpha n)$ bit flips.

(FelMalSerSteWai, ISIT-04)

Comments:

- key technical device: notice of dual witness for LP success
  - LP succeeds when 0^n sent $\iff$ primal optimum $p^* = 0$
  - suffices to construct dual optimal solution with $q^* = 0$
- caveat: constant fraction very low (e.g., $c = 0.00017$ for $R = 0.5$)
- potential gaps in the analysis
  - analysis adversarial in nature
  - dual witness relatively weak

Proof technique: Construction of dual witness

Primal LP: Vars. $\{\mu_i,\ i \in V\}$, $\{\mu_{a,J},\ a \in F, J \subseteq N(a),\quad |J| \text{ even}\}$

$$\text{min.}\quad \sum_{i \in V} \theta_i \mu_i \quad \text{s.t.} \begin{cases} \mu_{a,J} \geq 0 \\ \displaystyle\sum_{J \in \mathbb{C}(a)} \mu_{a,J} = 1 \\ \displaystyle\sum_{J \in \mathbb{C}(a), J_v = 1} \mu_{a,J} = \mu_v \end{cases}$$

Dual LP: Vars. $\{v_a,\ a \in F\}$ $\{\tau_{ia},\ (i,a) \in E\}$ unconstrained

$$\text{max.}\quad \sum_{a \in F} v_a \quad \text{s.t.} \begin{cases} \displaystyle\sum_{i \in S} \tau_{ia} \geq v_a \text{ for all} \quad a \in C, J \subseteq C(a), |J| \text{ even} \\ \displaystyle\sum_{a \in N(i)} \tau_{ia} \leq \theta_i \qquad \text{for all } i \in V \end{cases}$$

Dual witness to zero-valued primal solution

- assume WLOG that 0^n is sent: suffices to construct a dual solution with value $q^* = 0$
- dual LP simplifies substantially as follows:

Dual feasibility: Find real numbers $\{\tau_{ia},\ (i,a) \in E\}$ such that

$$\begin{aligned} \tau_{ia} + \tau_{ja} \quad & \geq \quad 0 \qquad \forall\, a \in C, \text{ and } i,j \in N(a) \\ \sum_{a \in N(i)} \tau_{ia} \quad & < \quad \theta_i \qquad \text{for all } i \in V \end{aligned}$$

- random weights $\theta_i \in \mathbb{R}$ defined by channel; e.g., for binary symmetric channel

$$\theta_i \quad = \quad \begin{cases} 1 & \text{with prob. } 1-p \\ -1 & \text{with prob. } p \end{cases}$$

Probabilistic analysis with random bit-flips

Consider an ensemble of LDPC codes with rate R, regular vertex degree d_v, and blocklength n. Suppose that the code is a $(\nu, \left(\frac{p}{d_v}\right) d_v)$ expander.

Theorem: For each (R, d_v, n), we specify fractions $\alpha > 0$ and error exponents $c > 0$ such that the LP decoder succeeds with probability $1 - \exp(-cn)$ over the space of bit flips $\leq \lfloor \alpha n \rfloor$. (DasDimKarWai07)

Remarks:

- the correctable fraction α is always larger than the worst case guarantee $\frac{3\frac{p}{d_v}-2}{2\frac{p}{d_v}-1}\nu$.
- concrete example: rate $R = 0.5$, degree $d_v = 8$ and $p = 6$ yields a correctable fraction $\alpha = 0.002$.

Hyperflow-based dual witness

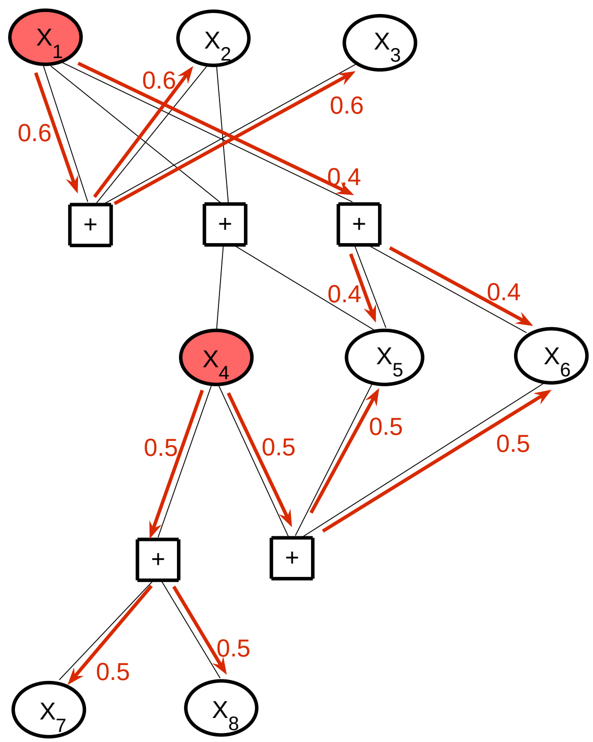
(DasDimKarWai07)

A *hyperflow* is a collection of weights $\{\tau_{ia}, (i,a) \in E\}$ such that:
(a) for each check $a \in F$, exists some $\gamma_a \geq 0$ and privileged neighbor $i^* \in N(a)$ such that

$$\tau_{ia} = \begin{cases} -\gamma_a & \text{for } i = i^* \\ +\gamma_a & \text{for } i \neq i^*. \end{cases}.$$

(b) $\sum_{a \in N(i)} \tau_{ia} < \theta_i$ for all $i \in V$.

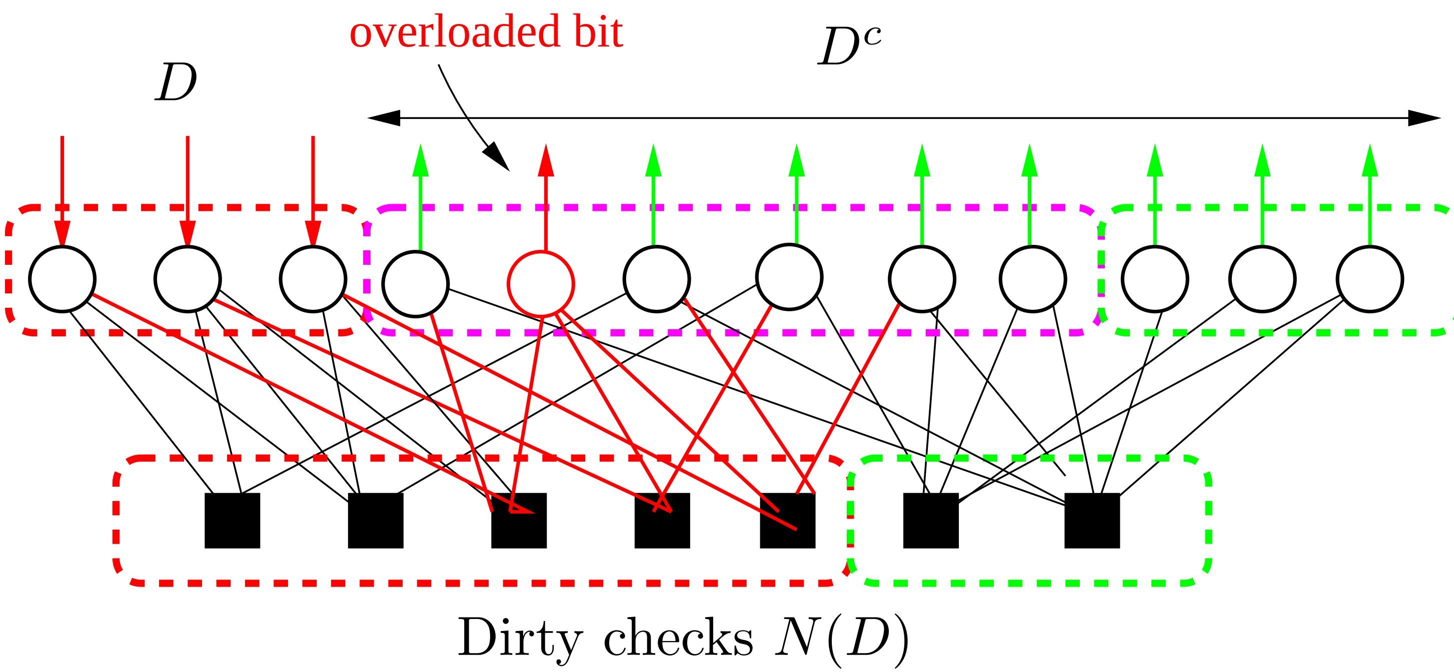
Proposition: A hyperflow exists $\Longleftrightarrow$ $\exists$ a dual feasible point with zero value.



Hyperflow (epidemic) interpretation:

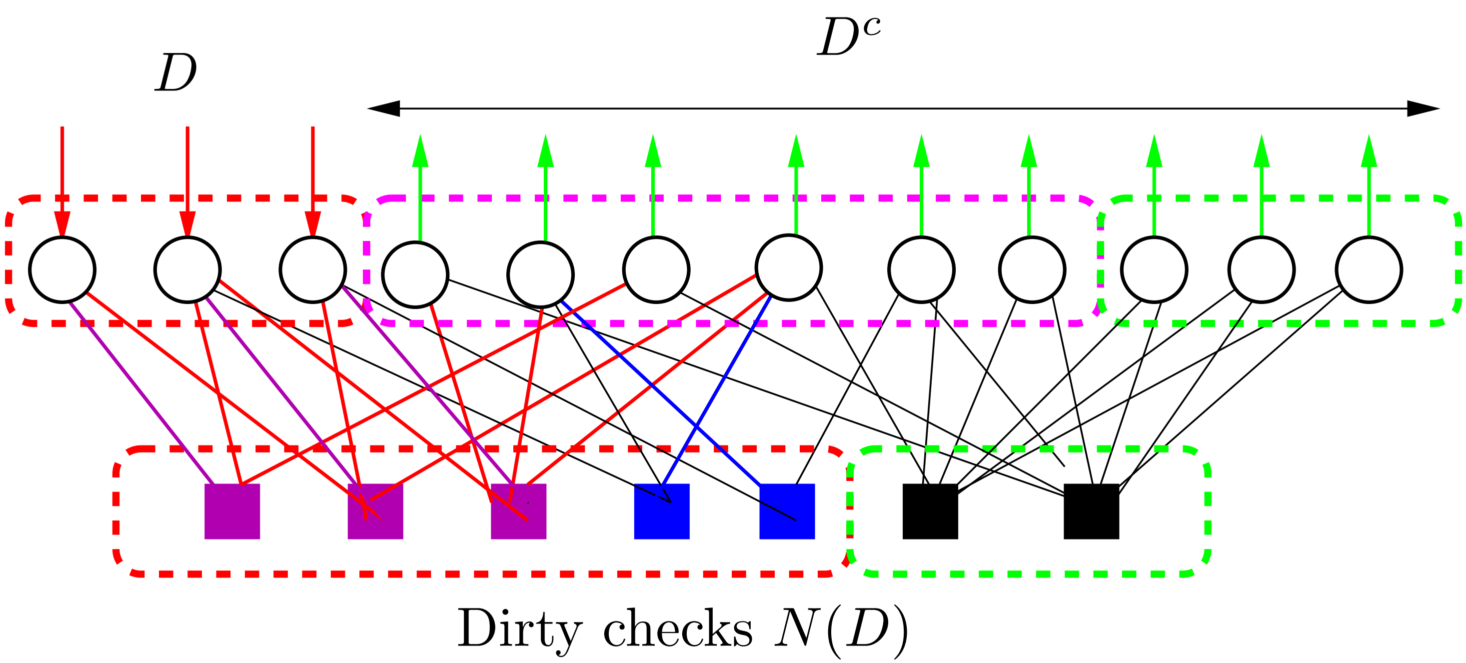
- each flipped bit adds 1 unit of "poison"; each clean bit absorbs at most 1 unit
- each infected check relays poison to all of its neighbors

Naive routing of poison may fail



- need to route 1 unit of poison away from each flipped bit
- each unflipped bit can neutralize at most one unit
- naive routing of poison can lead to overload

Routing poison via generalized matching



Definition: A (p, q)-matching is defined by the conditions:

(i) every flipped bit $i \in D$ is matched with p distinct checks.

(ii) every unflipped bit $j \in D^c$ matched with $\max\{Z_j - (d_v - q),\ 0\}$ checks from $N(D)$, where $Z_j = |N(j) \cap N(D)|$.

Generalized matching implies hyperflow

Lemma: Any (p, q) matching with $2p + q > 2d_v$ can be used to construct a valid hyperflow.

Proof:

- construct hyperflow with each flipped bit routing $\gamma \geq 0$ units to each of p checks
- each flipped bit can receive at most $(d_v - p)\gamma$ units from other dirty checks (to which it is not matched)
- hence we require that $-p\gamma + (d_v - p)\gamma < -1$, or $\gamma > 1/(2p - d_v)$
- each unflipped bit receives at most $(d_v - q)\gamma$ units so that we need $\gamma < 1/(d_v - q)$

High-level overview of key steps

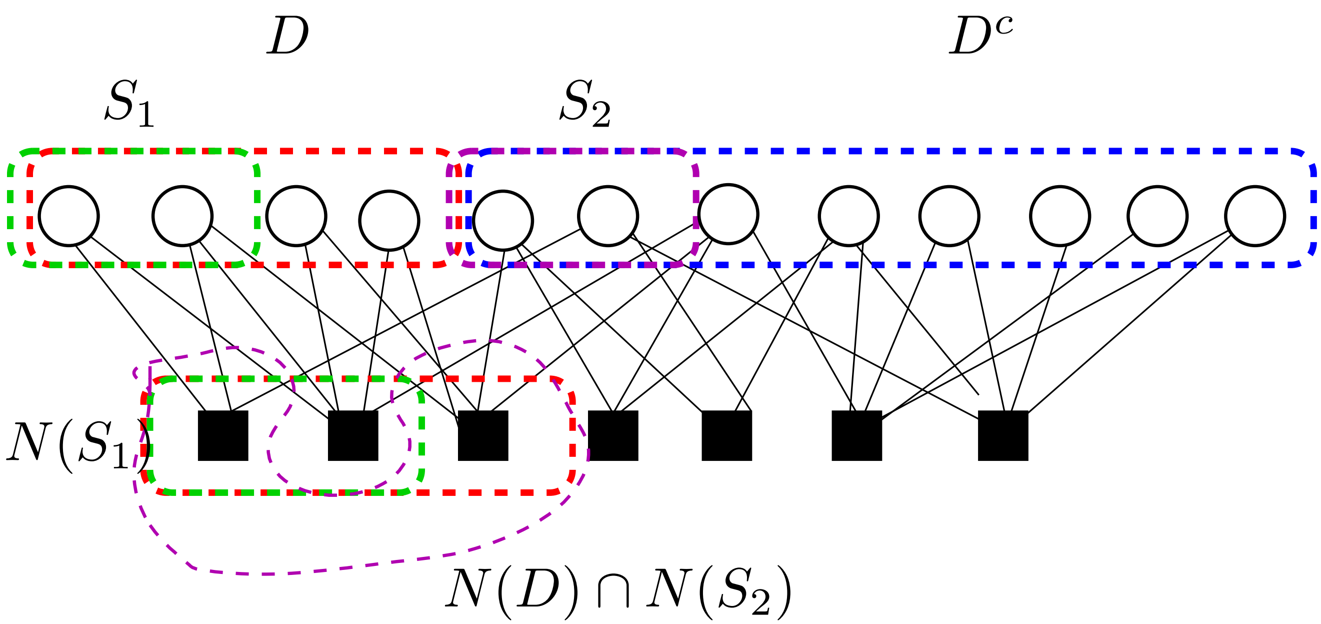
1. Randomly constructed LDPC is "almost-always" expander with high probability (w.h.p.)
   - weaker notion than classical expansion: holds for larger sizes
   - proof: union bounds plus martingale concentration

2. Prove that an "almost-always" expander will have a generalized matching w.h.p.
   - requires concentration statements
   - generalized Hall's theorem

3. Generalized matching guarantees existence of hyperflow.

4. Valid hyperflow is a dual witness for LP decoding succcess.

Generalized matching and Hall's theorem



- by generalized Hall's theorem, (p, q)-matching fails to exist if only if there exist subsets $S_1 \subseteq D$ and $S_2 \subseteq D^c$ that *contract*:

$$\underbrace{|N(S_1) \cup [N(S_2) \cap N(D)]|}_{\text{available matches}} \leq \underbrace{p|S_1| + \sum_{j \in S_2} \max\{0, q - (d_v - Z_j)\}}_{\text{total \# requests}}.$$

Analysis over a simpler random ensemble

- analysis in standard ensemble: complicated due to coupling between $N(D)$ and number of requests from D^c

- consider simplified (but equivalent) ensemble:
  - each node in D^c chooses $Z_j \sim \text{Bin}(d_v, \frac{|N(D)|}{m})$
  - chooses a subset from $N(D)$ of size Z_j

- LP error prob. (over random subset D) bounded by probability of existing contractive subsets $S_1 \subseteq D$ and $S_2 \subseteq D^c$:

$$\mathbb{P}\Big[\exists \quad S_1 \subseteq D, \quad S_2 \subseteq D^c \ \mid\ |N(S_1) \cup [N(S_2) \cap N(D)]| \leq p|S_1| + \sum_{j \in S_2} R_j\Big]$$

- argument establishes existence of "almost-always expanders" (with parameters much larger than worst-case sense)

Summary

- fruitful connections between two frameworks
  - message-passing in graphical models
  - LP relaxations for integer programs
- probabilistic analysis of LP relaxations
  - dual witness as certifiate of optimality
  - algorithmic correctness reduced to combinatorial analysis
- various open directions:
  - average-case analysis for other problems, ensembles?
  - guarantees on treewidth approximation hierarchies?

Related papers

1. M. J. Wainwright, T. Jaakkola and A. Willsky (2005). Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing. *IEEE Trans. Info. Theory* 51:11 pp. 3697–3717.

2. V. Kolmogorov and M. J. Wainwright (2005). On optimality properties of tree-reweighted max-product. *Proceedings of UAI.*

3. C. Daskalakis, A. G. Dimakis, R. M. Karp and M. J. Wainwright Probabilistic Analysis of Linear Programming Decoding. *IEEE Trans. Info. Theory*, To appear.

4. M. J. Wainwright and M. I. Jordan (2003). Graphical models, exponential families, and variational methods. UC Berkeley, Dept. of Statistics, Tech. Report 649.