# Optimization With Parity Constraints: From Binary Codes to Discrete Integration

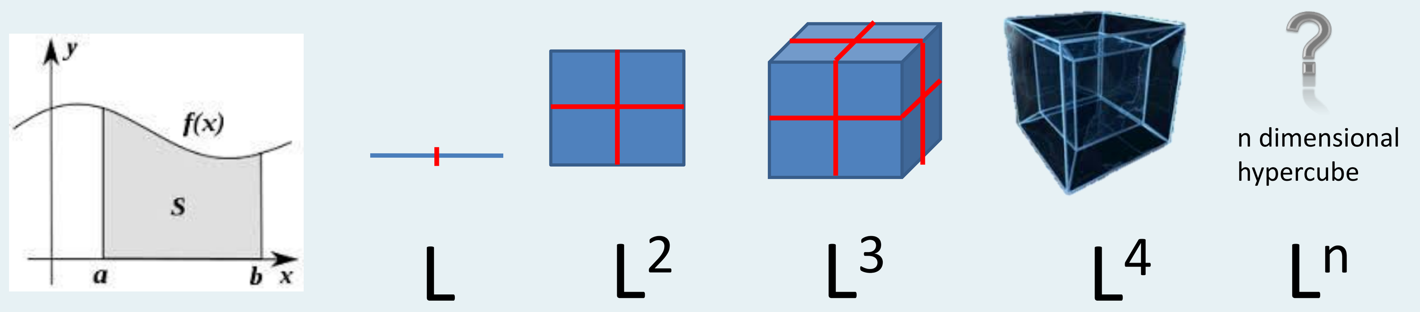

Stefano Ermon* Carla Gomes* Ashish Sabharwal+ Bart Selman*

*Computer Science Department, Cornell University, USA

+IBM Watson Research Center, NY, USA

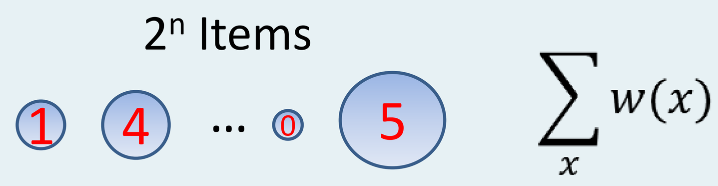

## Discrete Integration

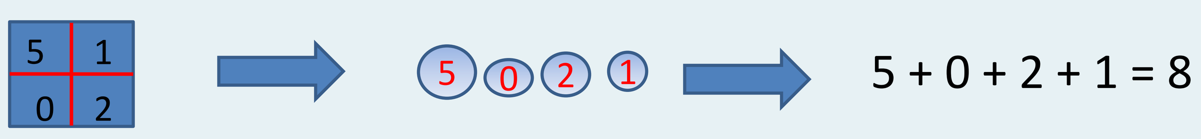
**The curse of dimensionality for high-dimensional integration**




**Discrete Integrals** ***(e.g., expectations, partition function, quadrature)***

- We are given
  - A set of $2^n$ items
  - Non-negative weights w
- Goal: compute **total weight**
- **Compactly specified** weight function (e.g, graphical model)



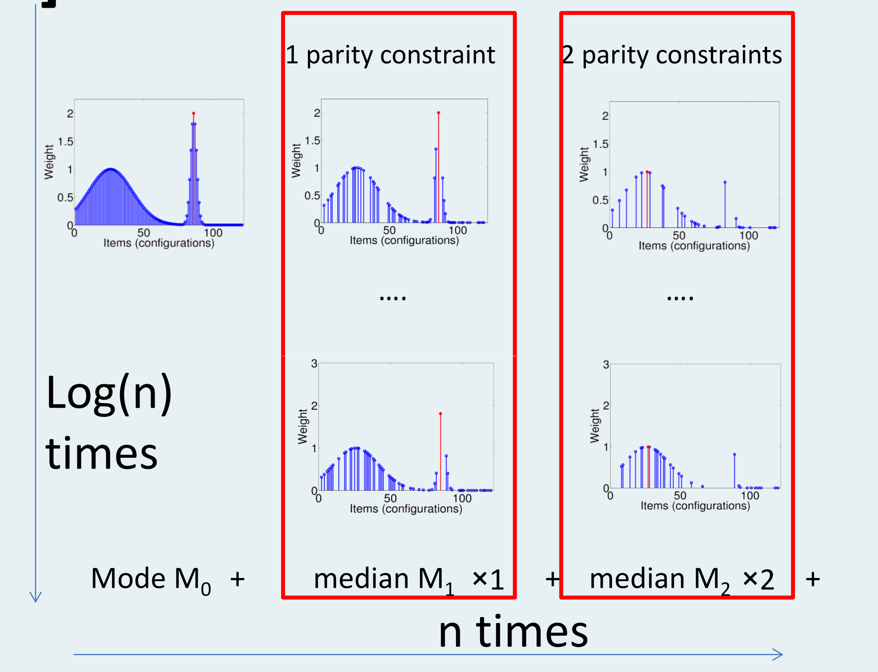




**Generally intractable** (e.g., 100 dimensions, sum over $2^{100} \sim 10^{30}$ items)

### WISH: Approximate Discrete Integration by Hashing and Optimization [ICML-13]

**Algorithm 1** WISH ($w : \Sigma \rightarrow \mathbb{R}^+, n = \log_2 |\Sigma|, \delta, \alpha$)

$T \leftarrow \left\lceil \frac{\ln(n/\delta)}{\alpha} \right\rceil$
**for** $i = 0, \cdots, n$ **do**
  **for** $t = 1, \cdots, T$ **do**
    Sample hash function $h^i_{A,b} : \Sigma \rightarrow \{0,1\}^i$, i.e.
      sample uniformly $A \in \{0,1\}^{i \times n}$, $b \in \{0,1\}^i$
    $w^t_i \leftarrow \max_\sigma w(\sigma)$ subject to $A\sigma = b \mod 2$
  **end for**
  $M_i \leftarrow \text{Median}(w^1_i, \cdots, w^T_i)$
**end for**
Return $M_0 + \sum_{i=0}^{n-1} M_{i+1} 2^i$




**1) Requires solving a small number of optimization instances (MAP)**

**2) (1+ ε) approximation with high probability (e.g., 99.9%)**

**3) Bounds on optimization → bounds on partition function**

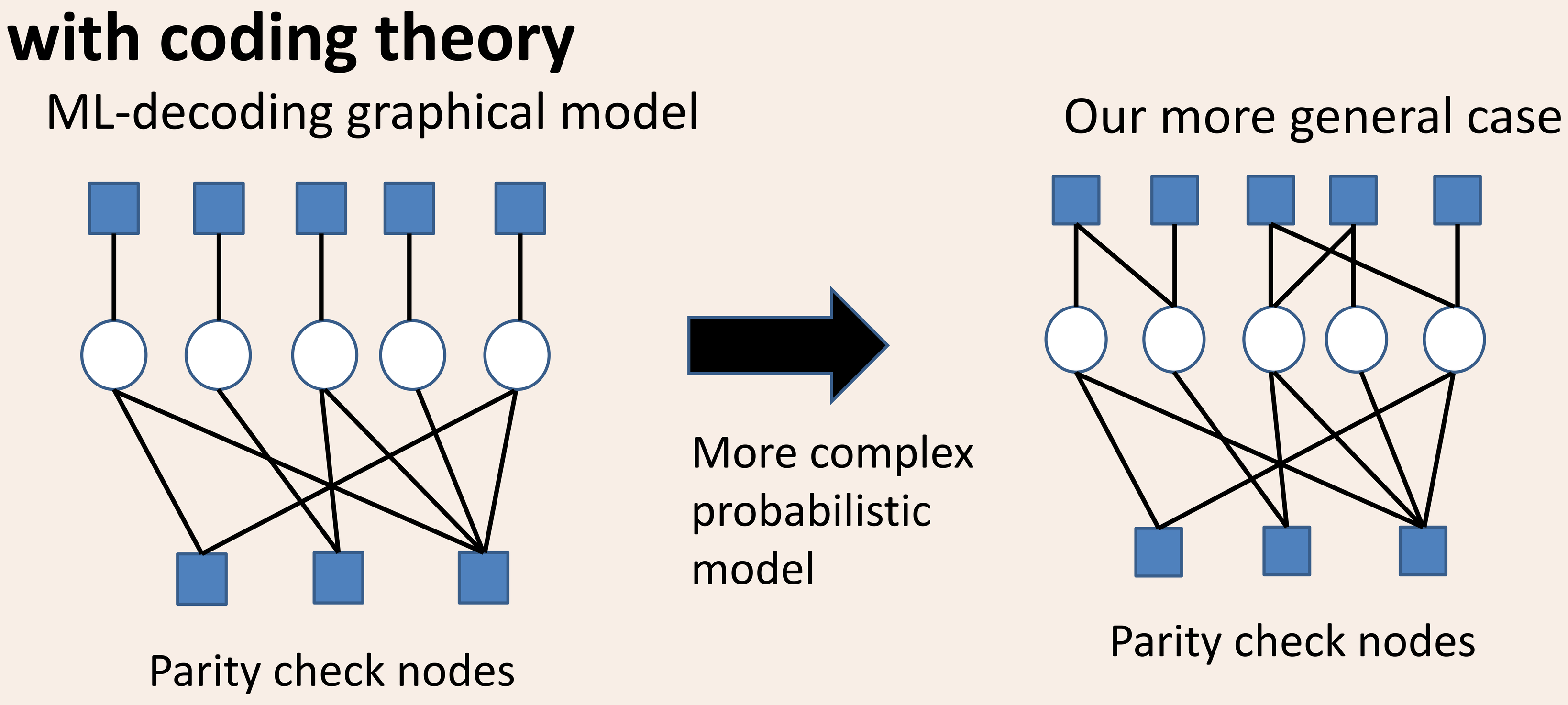
**Can we make it more scalable by approximating the optimizations in the inner loop?**

## Connections with coding theory

**Optimization in the inner loop is NP-hard (even to approximate within any constant factor)**

Reduction from:

**Definition 1** (MAXIMUM-LIKELIHOOD DECODING). Given a binary $m \times n$ matrix $A$, a vector $b \in \{0,1\}^m$, and an integer $w > 0$, is there a vector $z \in \{0,1\}^n$ of Hamming weight $\leq w$, such that $Az = b \mod 2$?



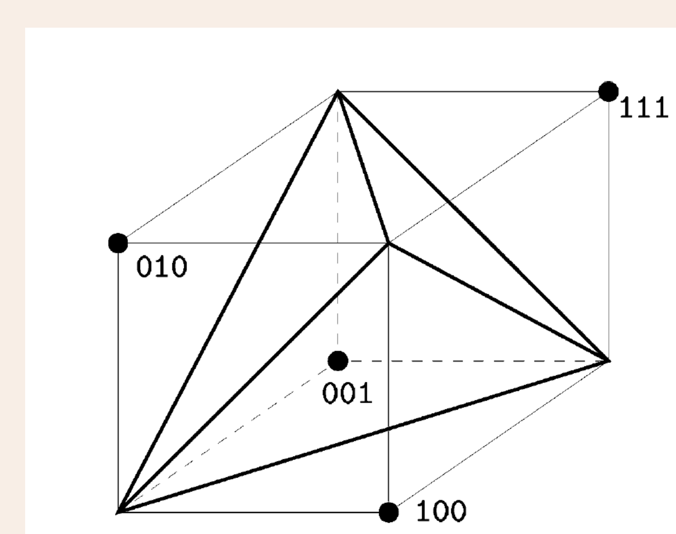

## Integer Linear Programming for MAP inference subject to parity constraints

**Formulate the NP-hard optimization max w(x) subject to A x = b (mod 2) as an Integer Linear Program**

- Effective strategy for decoding low-density parity check codes
- **Compact encoding** for parity constraints A x = b (mod 2) [Yannakakis,91]
- Upper and lower bounds

**1) LP relaxations provide polynomial time (probabilistic) upper bounds on the partition function**

**2) Branch and bound will eventually find an optimal integer solution (lower bound matches upper bound)**
**Provably within a constant factor of the true partition function** [ICML-13]




Parity polytope

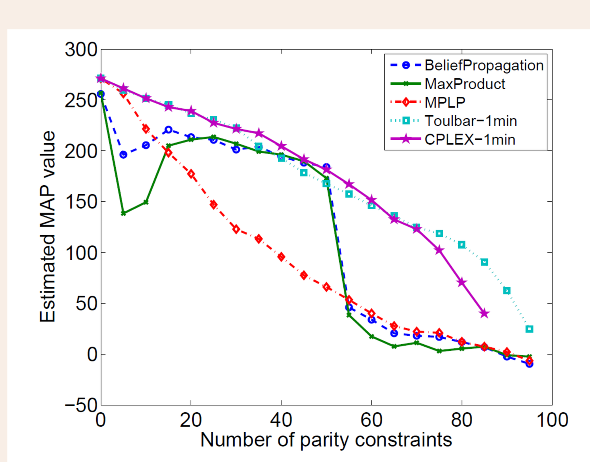
## Inducing sparsity to improve the relaxations

Problems with sparse A x = b are empirically easier to solve (similar to LDPC codes)

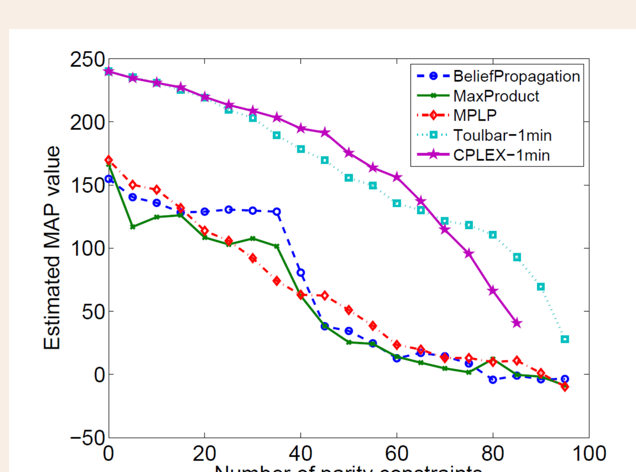
**1) Reduce A x = b to row-echelon form using Gauss-Jordan elimination**

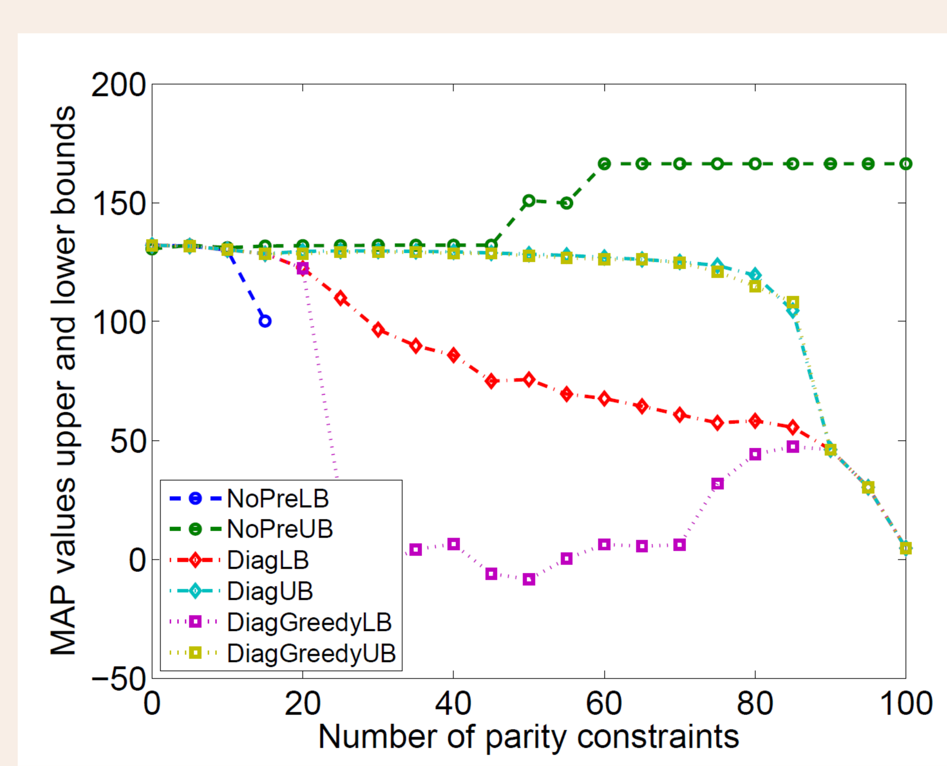
$$\begin{pmatrix} 1 & * & * & * & * & * & * & * & * \\ 0 & 1 & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 1 & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**2) Generate sparse matrices A. Still provides probabilistic lower bounds (but no upper bounds)**




(a) Attractive 10 × 10. Length 4 Xors

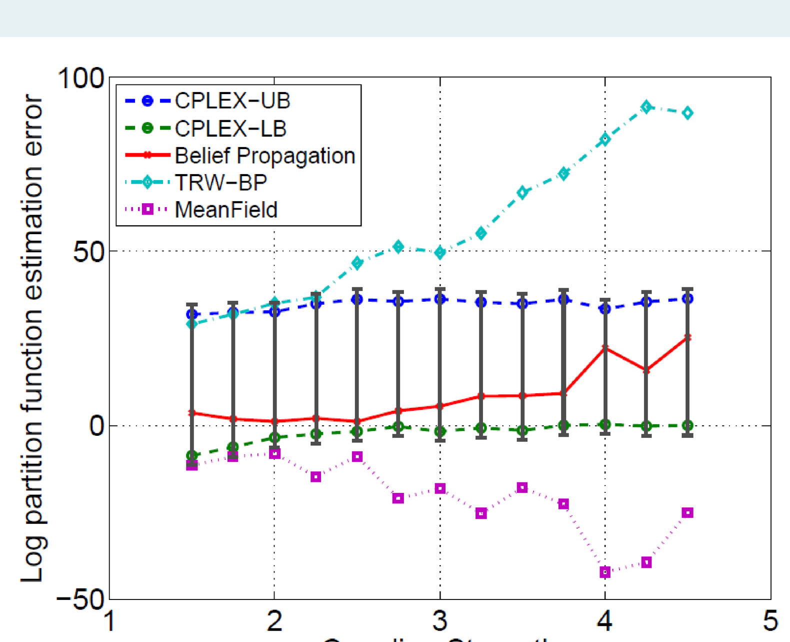


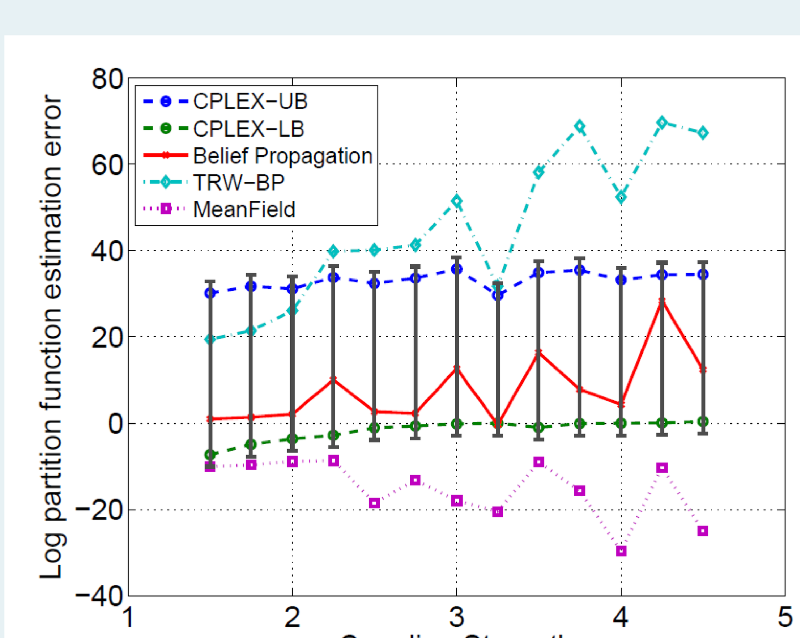

Improvements from sparsification

## Experimental results – Partition function via LP relaxations

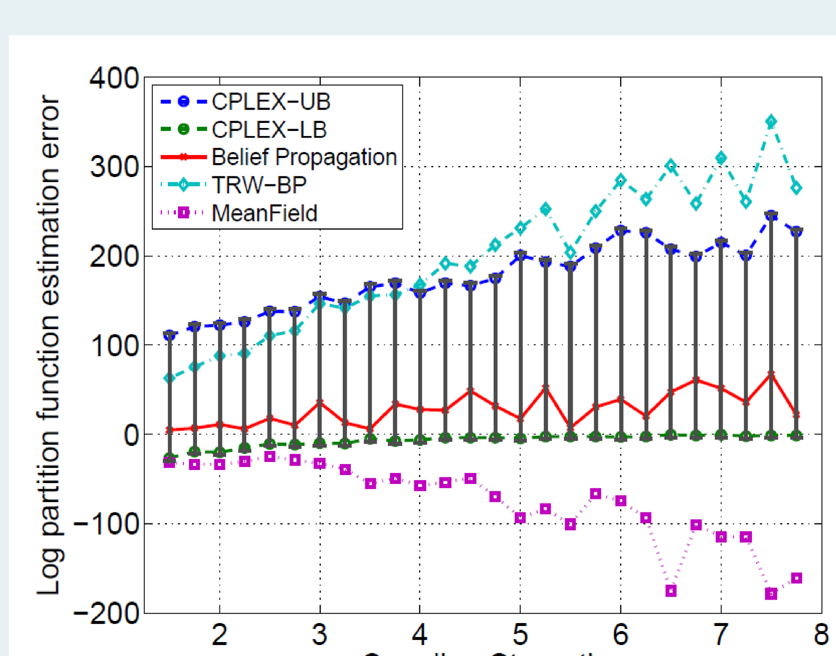
**ILP provides probabilistic upper and lower bounds that improve over time and are often tighter than variational methods**



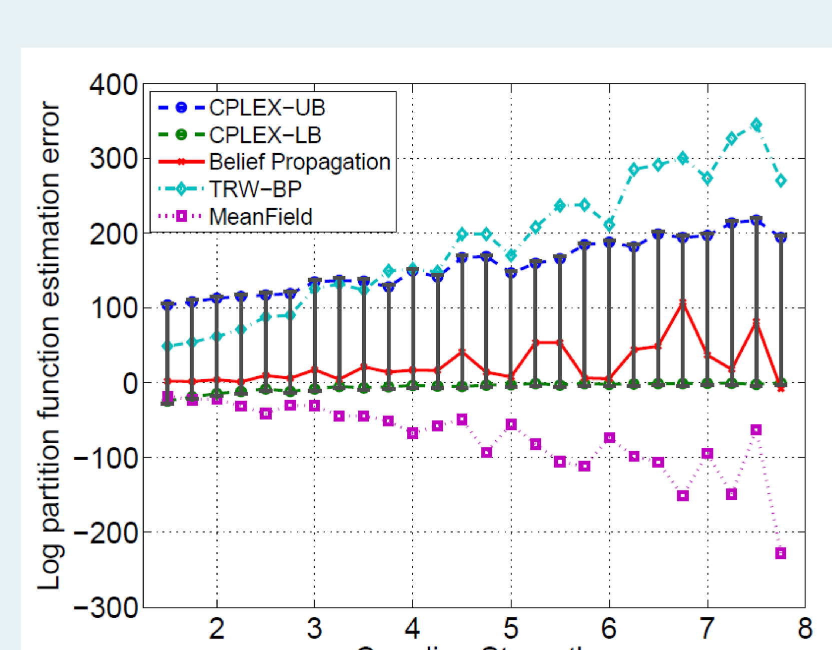

(c) Mixed 10 × 10. Field 0.1.




(d) Mixed 10 × 10. Field 1.0.




(e) Mixed 15 × 15. Field 0.1.




(f) Mixed 15 × 15. Field 1.0.