Markov Random Field MAP as Set Partitioning

James Cussens, University of York

PGM'18

Outline

The Set Partitioning Problem (SPP)

- If $V = \{1, 2, 3, 4\}$, and
- $E = \{\{1, 2\}, \{2, 3\}, \{1, 3\}, \{3, 4\}, \{2, 3, 4\}\}$ then
- $\{\{1, 2\}, \{3, 4\}\}$ is a set partition of the set V using E.

The Set Partitioning Problem (SPP)

- If $V = \{1, 2, 3, 4\}$, and
- $E = \{\{1, 2\}, \{2, 3\}, \{1, 3\}, \{3, 4\}, \{2, 3, 4\}\}$ then
- $\{\{1, 2\}, \{3, 4\}\}$ is a set partition of the set V using E.

- Let V be some finite set, and
- E be a set of subsets of V,
- so (V, E) is a hypergraph, where V is the set of vertices, and E the set of hyperedges.
- A partition of V is a set of subsets $E_1, \ldots, E_n$ such that:
  1. $\bigcup_i E_i = V$
  2. $E_i \cap E_j = \emptyset$, if $i \neq j$.
- So each element of V in exactly one E_i.

Integer programming formulation of SPP

- Let x_i be a binary variable where $x_i = 1$ states that the subset E_i is in the set partition.
- Each vertex must be in exactly one subset of the partition.
- So just need these constraints: $\forall v \in V : \sum_{i:v \in E_i} x_i = 1$ (*)
- If each subset E_i has some score c_i then we have the following integer program to find the best set partition:
- $\max \sum_i c_i x_i$ subject to constraints (*).
- Integer (linear) program since:
  - objective is linear and
  - all constraints are either linear or integrality constraints.

Set partitioning formally

A set partitioning (SPP) problem is of the following form:

$$\text{(SPP)} \qquad \min \{dx : Ax = e,\ x_j = 0 \text{ or } 1,\ \forall j \in N\}$$

where

- A is an $m \times n$ zero-one matrix,
- $d \in \mathbb{R}^n$ is an arbitrary objective,
- $e = (1, \ldots, 1)$ is an m-vector, and
- $N = \{1, \ldots, n\}$.

SPP example

If $V = \{1,2,3,4\}$, and $E = \{\{1,2\},\{2,3\},\{1,3\},\{3,4\},\{2,3,4\}\}$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The SPP linear relaxation (LSPP) and probability distributions

LSPP, the *linear relaxation* of an SPP problem, is formed by replacing the integrality constraints $x_i \in \{0, 1\}$ by the bounds $x_i \in [0, 1]$.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

- This LSPP states that the x_i have to define 4 ('overlapping') discrete probability distributions.
- The original SPP adds the additional constraint that each of the 4 probability distributions must be zero-one.
- This is basically why SPP is useful for MAP problems.

A Markov Random Field

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

$\times$

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

$\times$

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

$\times$

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

A Markov Random Field

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

$\times$

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

$\times$

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

$\times$

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

- $P(A=0, B=1, C=1, D=0) = (3 \times 5 \times 2 \times 1)/Z$
- where $Z = 495$, so
  $P(A=0, B=1, C=1, D=0) = 30/495 \approx 0.0606$
- The (single) MAP instantiation is $(A=0, B=1, C=1, D=1)$ with
  $P(A=0, B=1, C=1, D=1) = (3 \times 5 \times 2 \times 3)/495 = 0.\dot{1}\dot{8}$.
- Clearly $(A=0, B=1, C=1, D=1)$ also maximises $\log(Z \times P)$ with a value of $\log 3 + \log 5 + \log 2 + \log 3 \approx 4.49980$

MAP for MRFs formally

- The structure of the example MRF was the hypergraph with hyperedges $E = \{\{A, B\}, \{B, C\}, \{C, D\}, \{A, D\}\}$.
- *Any* hypergraph defines an MRF structure. The hyperedges do not have to correspond to the (maximal) clique of some graph.
- So MRF is just another word for *hierarchical log-linear model* [Lau96] .
- Just need to add *log potential functions* $\Psi = \{\psi_j : E_j \in E\}$.
- Let $x_{\downarrow j}$ denote some joint instantiation of the random variables in the hyperedge $E_j \in E$, then ψ_j maps each instantiation $x_{\downarrow j}$ to some value in $\mathbb{R} \cup \{-\infty\}$.
- The MAP problem for an MRF is to find a full joint instantiation x^* of maximal (log) probability: $x^* = \arg\max_x \sum_{E_j \in E} \psi_j(x_{\downarrow j})$

MAP for MRF as an Integer Program (IP)

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

×

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

×

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

×

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

Constraints: Marginal distributions are distributions

$$
\begin{aligned}
x(A_0B_0) + x(A_0B_1) + x(A_1B_0) + x(A_1B_1) &= 1 \\
x(B_0C_0) + x(B_0C_1) + x(B_1C_0) + x(B_1C_1) &= 1 \\
x(C_0D_0) + x(C_0D_1) + x(C_1D_0) + x(C_1D_1) &= 1 \\
x(A_0D_0) + x(A_0D_1) + x(A_1D_0) + x(A_1D_1) &= 1
\end{aligned}
$$

$$
\begin{array}{ll}
x(A_0) + x(A_1) = 1 & x(B_0) + x(B_1) = 1 \\
x(C_0) + x(C_1) = 1 & x(D_0) + x(D_1) = 1
\end{array}
$$

MAP for MRF as an Integer Program (IP)

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

×

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

×

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

×

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

Constraints: Marginal consistency

$$
\begin{aligned}
x(A_0) &= x(A_0 B_0) + x(A_0 B_1) = x(A_0 D_0) + x(A_0 D_1) \\
x(A_1) &= x(A_1 B_0) + x(A_1 B_1) = x(A_1 D_0) + x(A_1 D_1) \\
x(B_0) &= x(A_0 B_0) + x(A_1 B_0) = x(B_0 C_0) + x(B_0 C_1) \\
x(B_1) &= x(A_1 B_1) + x(A_1 B_1) = x(B_1 C_0) + x(C_1 C_1) \\
&\dots \\
x(D_1) &= x(A_0 D_1) + x(A_1 D_1) = x(C_0 D_1) + x(C_1 D_1)
\end{aligned}
$$

MAP for MRF as an Integer Program (IP)

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

×

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

×

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

×

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

Objective

$$\log(2)x(A_0B_0) + \log(3)x(A_0B_1) + \log(-\infty)\cancel{x(A_1B_0)} + \log(1)x(A_1B_1)$$
$$\log(5)x(B_0C_0) + \log(5)x(B_0C_1) + \log(5)x(B_1C_0) + \log(5)x(B_1C_1)$$
$$\log(4)x(C_0D_0) + \log(1)x(C_0D_1) + \log(2)x(C_1D_0) + \log(2)x(C_1D_1)$$
$$\log(1)x(A_0D_0) + \log(3)x(A_0D_1) + \log(1)x(A_1D_0) + \log(6)x(A_1D_1)$$

Reformulating as SPP

Change:

$$
\begin{aligned}
x(A_0) &= x(A_0B_0) + x(A_0B_1) = x(A_0D_0) + x(A_0D_1) \\
x(A_1) &= x(A_1B_0) + x(A_1B_1) = x(A_1D_0) + x(A_1D_1)
\end{aligned}
$$

to

$$
\begin{aligned}
x(A_1) + x(A_0B_0) + x(A_0B_1) &= 1 \\
x(A_1) + x(A_0D) + x(A_0D_1) &= 1
\end{aligned}
$$

etc

MAP for MRF as an Integer Program (IP)

- For each variable X_i and each value $x_i \in \mathcal{X}_i$ of X_i a binary IP variable $I(X_i = x_i)$, with zero objective coefficient, is created.
- For each hyperedge E_j and each *feasible* joint instantiation $x_{\downarrow j}$ (i.e. $\psi_j(x_{\downarrow j}) \neq -\infty$) of the variables in E_j, a binary IP variable $I(x_{\downarrow j})$ is created, with objective value $\psi_j(x_{\downarrow j})$.

Each random variable is assigned exactly one of its values:

$$\sum_{x_i \in \mathcal{X}_i} I(X_i = x_i) = 1$$

We also need marginal consistency for each random variable in each hyperedge:

$$\forall E_j, X_i \in E_j, x_i \in \mathcal{X}_i : I(X_i = x_i) = \sum_{x_{\downarrow j}(i) = x_i} I(x_{\downarrow j})$$

MAP for MRF as SPP

Replace

$$I(X_i = x_i) = \sum_{x_{\downarrow j}(i) = x_i} I(x_{\downarrow j})$$

with

$$\sum_{x_{i'} \in \mathcal{X}_i, x_i \neq x_{i'}} I(X_i = x_{i'}) + \sum_{x_{\downarrow j}(i) = x_i} I(x_{\downarrow j}) = 1$$

to get an SPP representation.

Tightening LSPP, the SPP linear relaxation

For each pair of hyperedges E_j, $E_{j'}$ whose intersection contains at least 2 random variables and for each instantiation $x_{j \cap j'}$ of the random variables in $E_j \cap E_{j'}$ we **may** add a constraint that E_j and $E_{j'}$ must be consistent for $x_{j \cap j'}$:

$$\sum_{x_{\downarrow j}(j \cap j') = x_{j \cap j'}} I(x_{\downarrow j}) = \sum_{x_{\downarrow j'}(j \cap j') = x_{j \cap j'}} I(x_{\downarrow j'})$$

- These constrain LSPP (but not SPP) giving us better bounds from solving LSPP (at the expense of more constraints).
- Constraints of this sort are given, for example, by [KF09, §13.5.1] in their presentation, but were not used by [HOA$^+$16].

Bounds from LSPP

Solving the linear relaxation of the MAP problem (LSPP) for this MRF results in an integer solution:

A	B	
0	0	2
0	1	3
1	0	0
1	1	1

$\times$

B	C	
0	0	5
0	1	5
1	0	5
1	1	5

$\times$

C	D	
0	0	4
0	1	1
1	0	2
1	1	2

$\times$

A	D	
0	0	1
0	1	3
1	0	1
1	1	6

- So in this case solving LSPP solves SPP.
- Usually we are not so lucky!

A different objective

We solve the LSPP and SPP for this MRF MAP instance:

A	B	
0	0	5
0	1	1
1	0	1
1	1	1

$\times$

B	C	
0	0	1
0	1	1
1	0	5
1	1	1

$\times$

C	D	
0	0	1
0	1	1
1	0	5
1	1	1

$\times$

A	D	
0	0	1
0	1	1
1	0	5
1	1	5

A fractional solution to LSPP

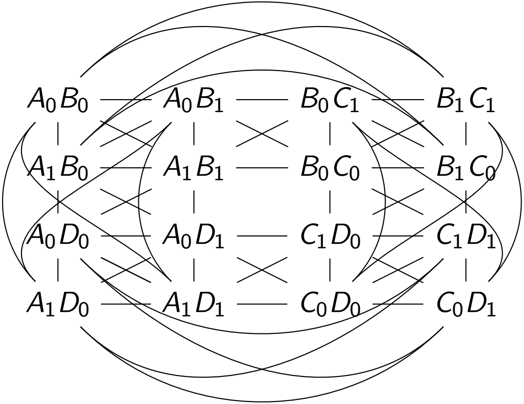
The solution to the LSPP is:

objective value:	4.02359478108525
x$A0B0	0.5
x$A1B1	0.5
x$B0C1	0.5
x$B1C0	0.5
x$C0D1	0.5
x$C1D0	0.5
x$A0D1	0.5
x$A1D0	0.5

whereas the solution to the **SPP** is

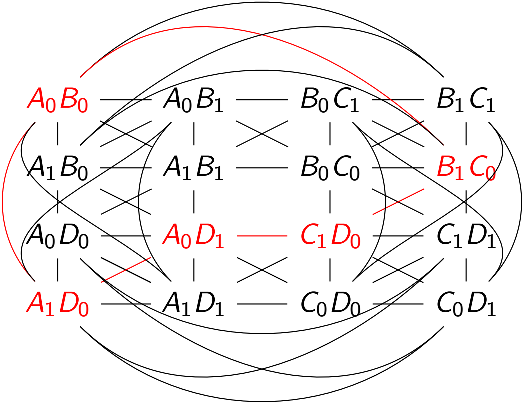
objective value:	3.2188758248682
x$A0b1	1
x$B1C0	1
x$C0D1	1
x$A0D1	1

The SPP intersection graph



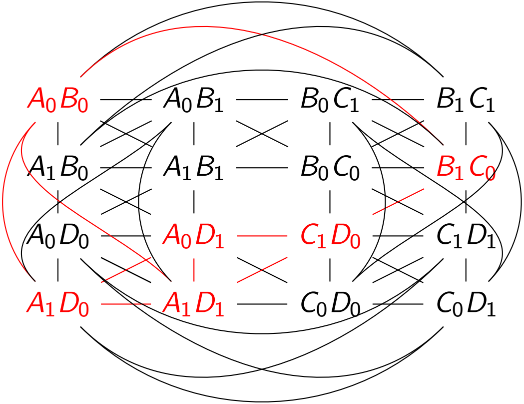
The intersection graph for $\{\{A, B\}, \{B, C\}, \{C, D\}, \{A, D\}\}$

The SPP intersection graph



This odd hole in the intersection graph determines this *cut*:
$A_0B_0 + B_1C_0 + C_1D_0 + A_0D_1 + A_1D_0 \leq 2$

The SPP intersection graph



The cut can be 'lifted':

$A_0 B_0 + B_1 C_0 + C_1 D_0 + A_0 D_1 + A_1 D_0 + A_1 D_1 \leq 2$

Tightening the linear relaxation by adding cutting planes

After adding the odd-hole cut:

```
objective value:                      3.2188758248682
x$A0B0                              0.333333333333333
x$A1B1                              0.666666666666667
x$B0C1                              0.333333333333333
x$B1C0                              0.666666666666667
x$C0D1                              0.666666666666667
x$C1D0                              0.333333333333333
x$A0D1                              0.333333333333333
x$A1D0                              0.333333333333333
x$A1D1                              0.333333333333333
```

An integer solution after adding the lifted odd-hole cut!

```
objective value:                      3.2188758248682
x$A0B0                                              1
x$B0C1                                              1
x$C1D1                                              1
x$A0D1                                              1
```

Anti-holes and perfection

- An *anti-hole* is the complement of a hole.
- We can also generate cuts from odd anti-holes ...
- ... but time is short!
- If the intersection graph contains no odd holes and no odd anti-holes it is *perfect* and the LSPP will always result in an integer solution.

Zero-half cuts

- Ideally we would have a specialised cutting plane algorithm to look for lifted odd-hole / anti-hole cuts.
- The lazy option is to use an off-the-shelf solver—in my case CPLEX—and encourage it to look for *zero-half cuts*.
- Odd hole / anti-hole cuts are zero-half cuts (see paper).

$$
\begin{array}{rll}
 & (x_1 + x_2 \leq 1)/2 & \\
+ & (x_1 + x_3 \leq 1)/2 & \\
+ & (x_2 + x_3 \leq 1)/2 & \\
= & x_1 + x_2 + x_3 \leq 3/2 & \\
\Rightarrow & x_1 + x_2 + x_3 \leq 1 & \qquad\qquad \text{if } x_i \text{ integer}
\end{array}
$$

Generating many zero-half cuts with CPLEX

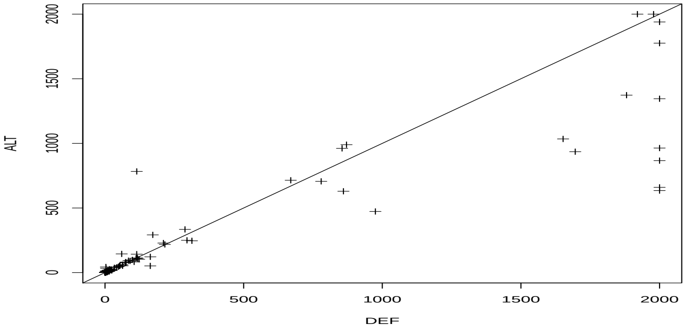


Figure: CPLEX solving times. Many zero-half cuts (ALT) vs default parameters (DEF) on 320 MRF MAP instances. 2000 seconds is timeout.

Removing redundant hyperedges

If $E_j \subseteq E_{j'}$ then E_j is *redundant* and removed from E. Its log potential function is added to that of $E_{j'}$:

$$\psi_{j'} \leftarrow \psi_{j'} + \psi_j$$

Removing random variables in only one hyperedge

- If a random variable occurs in only one hyperedge, then this hyperedge is its Markov blanket.
- Just record its ‘best’ value for each instantiation of the Markov blanket and remove it from the hyperedge (and thus from the problem).
- This may produce a redundant hyperedge.
- Continually removing variables in only one hyperedge and removing redundant hyperedges is *Graham’s algorithm*.
- Iff the hypergraph is *decomposable* then Graham’s algorithm is enough to solve the problem.

Removing a random variables in several hyperedges

- *Any* two hyperedges E_j and $E_{j'}$ can be removed from the MRF and replaced by their union $E_{j''} = E_j \cup E_{j'}$ where $\psi_{j''} \leftarrow \psi_j + \psi_{j'}$
- It follows that any variable X_i can be removed from the MRF by:
  1. replacing the hyperedges which contain it with the single hyperedge $E_{j(i)}$, which is their union,
  2. eliminating X_i (now contained only in $E_{j(i)}$) using the procedure given earlier, and
  3. removing any newly redundant hyperedges
- Apply only when this does not increase the number of IP variables.
- Generalisation: Can also be non-increasing even if it is not possible to eliminate any random variables. Destroy that Markov structure!
- (This is basically 'Sherali-Adams'.)

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.
- Can do similarly for the other 15 possible instantiations of B, C, D, E.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.
- Can do similarly for the other 15 possible instantiations of B, C, D, E.
- Exactly one of these 16 products will $= 1$ in any optimal solution, the others will $= 0$.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.
- Can do similarly for the other 15 possible instantiations of B, C, D, E.
- Exactly one of these 16 products will $= 1$ in any optimal solution, the others will $= 0$.
- If, say, $x(a_0 b_1 c_0)$ never appears in the 16 products it must be 0 in any optimal solution, so we can simply set it to zero.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.
- Can do similarly for the other 15 possible instantiations of B, C, D, E.
- Exactly one of these 16 products will $= 1$ in any optimal solution, the others will $= 0$.
- If, say, $x(a_0 b_1 c_0)$ never appears in the 16 products it must be 0 in any optimal solution, so we can simply set it to zero.
- This 'domain reduction' can 'propagate' so we apply it in rounds until no further progress is made.

Using Markov structure

- Suppose the Markov blanket for A is $E_1 = \{A, B, C\}$, $E_2 = \{A, B, D\}$ and $E_3 = \{A, E\}$, and all 5 variables are binary.
- Suppose $A = 1$ is the optimal choice for variable A when $B = C = D = E = 0$.
- So if $B = C = D = E = 0$, then $x(a_1 b_0 c_0)x(a_1 b_0 d_0)x(a_1 e_0) = 1$ in any optimal solution.
- Can do similarly for the other 15 possible instantiations of B, C, D, E.
- Exactly one of these 16 products will $= 1$ in any optimal solution, the others will $= 0$.
- If, say, $x(a_0 b_1 c_0)$ never appears in the 16 products it must be 0 in any optimal solution, so we can simply set it to zero.
- This 'domain reduction' can 'propagate' so we apply it in rounds until no further progress is made.

Reducing an MRF

```
Hyperedges=19200,RVs=6400,HypInsts=64000 originally
Hyperedges=12800,RVs=6400,HypInsts=51200 after using evidence and reducing
Hyperedges=12800,RVs=6400,HypInsts=51200 after running Graham's algorithm
Hyperedges=3200,RVs=3200,HypInsts=51200 after eliminating variables with 'small Mar
Hyperedges=3200,RVs=3200,HypInsts=51200 after making products
Hyperedges=3200,RVs=3200,HypInsts=39224 after removing suboptimal insts
Hyperedges=3200,RVs=3200,HypInsts=39224 after running Graham's algorithm
Hyperedges=3200,RVs=3200,HypInsts=39224 after eliminating variables with 'small Mar
Hyperedges=3124,RVs=3200,HypInsts=39146 after making products
Hyperedges=3124,RVs=3200,HypInsts=39096 after removing suboptimal insts
Hyperedges=3124,RVs=3200,HypInsts=39096 after running Graham's algorithm
Hyperedges=3124,RVs=3200,HypInsts=39096 after eliminating variables with 'small Mar
Hyperedges=3124,RVs=3200,HypInsts=39096 after making products
Hyperedges=3124,RVs=3200,HypInsts=39096 after removing suboptimal insts
Making names for hyperedge instantiations ... Done
Created 39096 MIP variables for hyperedge instantiations
Created 6400 'I(v=val)' MIP variables and 3200 set partitioning constraints MRF var
Created 45496 MIP variables in total
Added 3124 normal marginal consistency constraints
Added 0 using-default marginal consistency constraints
Solving...
```

CPLEX solving a reduced MRF

```
MIP Presolve eliminated 25836 rows and 1 columns.
MIP Presolve eliminated 53 rows and 38 columns.
Reduced MIP has 40233 rows, 41975 columns, and 272162 nonzeros.
Reduced MIP has 41975 binaries, 0 generals, 0 SOSs, and 0 indicators.

        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                     -1700944.0000       -0.0000    17615  100.00%
      0     0  -949740.0000  1781 -1700944.0000  -949740.0000    17615   44.16%
*     0+    0                      -997147.0000  -949740.0000    17615    4.75%
      0     0  -950254.0000   876  -997147.0000      Cuts: 96    17744    4.70%
*     0+    0                      -977043.0000  -950254.0000    17744    2.74%
      0     0  -950377.2500  1589  -977043.0000  ZeroHalf: 18    17796    2.73%
*     0+    0                      -954453.0000  -950377.2500    17796    0.43%
      0     0  -950425.0000  1328  -954453.0000  ZeroHalf: 40    17800    0.42%
*     0+    0                      -950425.0000  -950425.0000    17800    0.00%
      0     0        cutoff        -950425.0000  -950425.0000    17800    0.00%
Elapsed time = 4.21 sec. (3409.94 ticks, tree = 0.00 MB, solutions = 5)

Zero-half cuts applied:  108.  Lift and project cuts applied:  1
Gomory fractional cuts applied:  2
```

Applying problem reduction on 'grids' instances

		CPLEX(ZRI)		HURLEY-BEST	
Instance	OPT	RED=NO	RED=YES	OBJ	t
40.10	370567	16	39(24)	370567	101
40.10.w	398635	20	16(1)	398635	457
40.15	521289	19	47(27)	521289	149
40.15.w	562547	27	20(1)	562547	546
80.10	1555353	1121	1643(1556)	1558819	3600
80.10.w	1632909	3071	87(12)	1646415	3600
80.15	2190818	2150	*3678(3600)	2239504	3600
80.15.w	2272095	**3600	79(13)	2293410	3600
80.2	511424	18	117(13)	511424	121
80.2.w	514152	20	77(2)	514152	110
80.5	917776	71	151(67)	917946	3600
80.5.w	950425	152	68(4)	951943	3600

But on the other hand ...

- Comparing against Hurley *et al* [HOA+16] on other difficult instances (UAI PIC competition ones), the SPP approach (with or without reduction) does badly.
- Many of these problems have a hyperedge for every pair of random variables. No Markov structure.

After an hour of cutting . . .

```
      Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                      -765832.0000       -0.0000    10353  100.00%
      0     0  -636094.0000  4149  -765832.0000  -636094.0000    10353   16.94%
      0     0  -637140.0000  4510  -765832.0000    Cuts: 1209    11467   16.80%
      0     0  -638091.3289  5269  -765832.0000    Cuts: 1445    13022   16.68%
      0     0  -639059.7545  5870  -765832.0000 ZeroHalf: 1492    14831   16.55%
      0     0  -639578.2045  6028  -765832.0000 ZeroHalf: 1500    16083   16.49%
      0     0  -639896.5485  6676  -765832.0000 ZeroHalf: 1400    17730   16.44%
      0     0  -640358.2798  6674  -765832.0000 ZeroHalf: 1390    19924   16.38%
      0     0  -640753.0583  7451  -765832.0000 ZeroHalf: 1418    21746   16.33%
...
      0     0  -652767.8813 11817  -765832.0000  ZeroHalf: 20  1728422   14.76%
      0     0  -652769.2701 11831  -765832.0000  ZeroHalf: 36  1729801   14.76%
      0     0  -652770.9797 11825  -765832.0000  ZeroHalf: 37  1731359   14.76%

Zero-half cuts applied:  2754
```

Related work

- The SPP polytope (convex hull of solutions) for MRFs is nothing other than the *marginal polytope* [WJ03].
- Considerable existing work on using LP relaxations of this polytope [SJ08, BNK11, SCL12, MTGW13, HOA+16, RPW17].
- Including work on cycle inequalities [SCL12] and 'frustrated cycles' [SJ08].
- This is to say nothing of the very large literature on set partitioning (and related problems).

Future directions

- One must strike the right balance between getting a better bound and branching.
- Many of the reformulation methods should be applied *throughout the search*, e.g. as propagators.
- Or more radically: exploit the fact that the SPP polytope (= marginal polytope) is *quasi-integral*.
- This means "the set partitioning problem can in principle be solved by a modified version of the simplex method, generating only integer solutions" [BP76].

Dhruv Batra, Sebastian Nowozin, and Pushmeet Kohli.
Tighter relaxations for MAP-MRF inference: A local primal-dual gap based separation algorithm.
In *Proc. Artificial Intelligence and Statistics*, 2011.

Egon Balas and Manfred W. Padberg.
Set partitioning: A survey.
SIAM Review, 18(4):710–760, October 1976.

Barry Hurley, Barry O'Sullivan, David Allouche, George Katsirelos, Thomas Schiex, Matthias Zytnicki, and Simon De Givry.
Multi-language evaluation of exact solvers in graphical model discrete optimization.
Constraints, 21(3):413–434, July 2016.

Daphne Koller and Nir Friedman.
Probabilistic Graphical Models: Principles and Techniques.
MIT Press, 2009.

Steffen L. Lauritzen.

Graphical Models.
Oxford University Press, Oxford, 1996.

Elad Mezuman, Daniel Tarlow, Amir Globerson, and Yair Weiss.
Tighter linear program relaxations for high order graphical models.
In *Proc. UAI 2013*, 2013.

M. Rowland, A. Pacchiano, and A. Weller.
Conditions beyond treewidth for tightness of higher-order LP relaxations.
In *Proc. Artificial Intelligence and Statistics (AISTATS)*, 2017.

David Sontag, Do Kook Choe, and Yitao Li.
Efficiently searching for frustrated cycles in MAP inference.
In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pages 795–804, Corvallis, Oregon, 2012. AUAI Press.

David Sontag and Tommi Jaakkola.
New outer bounds on the marginal polytope.

In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1393–1400, Cambridge, MA, 2008. MIT Press.

M. Wainwright and M. I. Jordan.
Graphical models, exponential families and variational inference.
Technical Report 649, UC Berkeley, Dept. of Statistics, 2003.