Graphical Models

Logic, costs, & probabilities

Nesy-Montreal

T. Schiex

Université Fédérale de Toulouse, ANITI, INRAE MIAT, Toulouse, France

May 5, 2025

Graphical models [16, 30, 20, 37]

The definition of a multivariate function as the *combination* of *simple* functions.

- Constraint networks, SAT (CNF), CP: boolean, feasibility
- Cost Function Networks, Max-SAT, COP: cost and feasibility
- Markov Random Fields, Bayesian Nets: probabilities (normalization)
- Algebraic frameworks (Valued/monoïds [32], Semi-Ring [36, 8, 24]),…

Graphical models [16, 30, 20, 37]

The definition of a multivariate function as the *combination* of *simple* functions.

- Constraint networks, SAT (CNF), CP: boolean, feasibility
- Cost Function Networks, Max-SAT, COP: cost and feasibility
- Markov Random Fields, Bayesian Nets: probabilities (normalization)
- Algebraic frameworks (Valued/monoïds [32], Semi-Ring [36, 8, 24]),...

Graphical models [16, 30, 20, 37]

The definition of a multivariate function as the *combination* of *simple* functions.

- Constraint networks, SAT (CNF), CP: boolean, feasibility
- Cost Function Networks, Max-SAT, COP: cost and feasibility
- Markov Random Fields, Bayesian Nets: probabilities (normalization)
- Algebraic frameworks (Valued/monoïds [32], Semi-Ring [36, 8, 24]),...

Graphical models [16, 30, 20, 37]

The definition of a multivariate function as the *combination* of *simple* functions.

- Constraint networks, SAT (CNF), CP: boolean, feasibility
- Cost Function Networks, Max-SAT, COP: cost and feasibility
- Markov Random Fields, Bayesian Nets: probabilities (normalization)
- Algebraic frameworks (Valued/monoïds [32], Semi-Ring [36, 8, 24]),...

ANITI INRAE

A Constraint Network $\langle \boldsymbol{V}, \Phi \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e Boolean functions (or constraints)
- Each $\varphi_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

$$\Phi_{\mathcal{M}} = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Constraint Satisfaction Problem (NP-complete)

- Is it possible to have $\Phi_{\mathcal{M}} = t$?

GMs for feasibility (combination operator)

A Constraint Network $\langle \boldsymbol{V}, \Phi \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e Boolean functions (or constraints)
- Each $\varphi_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \rightarrow \{t, f\}$

Joint truth function

$$\Phi_{\mathcal{M}} = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Constraint Satisfaction Problem (NP-complete)

- Is it possible to have $\Phi_{\mathcal{M}} = t$?

GMs for feasibility (combination operator)

A Constraint Network $\langle \boldsymbol{V}, \Phi \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e Boolean functions (or constraints)
- Each $\varphi_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

$$\Phi_{\mathcal{M}} = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Constraint Satisfaction Problem (NP-complete)

- Is it possible to have $\Phi_{\mathcal{M}} = t$?

ANITI INRAE

A Constraint Network $\langle \boldsymbol{V}, \Phi \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e Boolean functions (or constraints)
- Each $\varphi_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

$$\Phi_{\mathcal{M}} = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Constraint Satisfaction Problem (NP-complete)

- Is it possible to have $\Phi_{\mathcal{M}} = t$?

A Constraint Network $\langle \boldsymbol{V}, \Phi \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e Boolean functions (or constraints)
- Each $\varphi_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

$$\Phi_{\mathcal{M}} = \bigwedge_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Constraint Satisfaction Problem (NP-complete)

- Is it possible to have $\Phi_{\mathcal{M}} = t$?

Languages for domains and constraints

- Constraint Networks: Boolean tables (tensors) for domains and constraints
- SAT/CNF: Boolean variables and clauses
- Constraint Programming: interval domains, specialized constraints

Tables (or tensors) for φ_S

- A multidimensional table with a Boolean for every tuple in D^S
- Says if it is authorized (t) or not (f)

Pairwise difference (3 values)

$$\begin{bmatrix} f & t & t \\ t & f & t \\ t & t & f \end{bmatrix}$$

Languages for domains and constraints

- Constraint Networks: Boolean tables (tensors) for domains and constraints
- SAT/CNF: Boolean variables and clauses
- Constraint Programming: interval domains, specialized constraints

Beyond dense tensors

- Clauses: forbids one cell of a (sparse) tensor
- Names for specific (useful) constraints

GMs for feasibility (language): CN, SAT, CP

Languages for domains and constraints

- Constraint Networks: Boolean tables (tensors) for domains and constraints
- SAT/CNF: Boolean variables and clauses
- Constraint Programming: interval domains, specialized constraints

Beyond dense tensors

- Clauses: forbids one cell of a (sparse) tensor
- Names for specific (useful) constraints

Most famous

AllDifferent$s$

ANITI INRAE

Languages for domains and constraints

- Constraint Networks: Boolean tables (tensors) for domains and constraints
- SAT/CNF: Boolean variables and clauses
- Constraint Programming: interval domains, specialized constraints

Application domains: NP and beyond

Excel at the analysis of complex perfectly known systems

Digital circuit verification, scheduling and other resource management problems, planning, software verification, theorem proving,…

Biology?

ANITI INRAE

Cost Function Network $\langle \boldsymbol{V}, \Phi, k\rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e integer cost functions
- each $\varphi_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by k (finite or infinite)

Joint cost function using $a +^k b = \min(a+b, k)$ saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum^{k}_{\varphi_{\boldsymbol{S}} \in \Phi} \varphi_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

- What/where is the minimum of $\Phi_{\mathcal{M}}$?

Cost Function Network $\langle \boldsymbol{V}, \Phi, k \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e integer cost functions
- each $\varphi_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by k (finite or infinite)

Joint cost function using $a +^k b = \min(a + b, k)$ saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{\varphi_{\boldsymbol{S}} \in \Phi}^{k} \varphi_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

- What/where is the minimum of $\Phi_{\mathcal{M}}$?

ANITI INRAE

Cost Function Network $\langle \boldsymbol{V}, \Phi, k \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e integer cost functions
- each $\varphi_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by k (finite or infinite)

Joint cost function using $a +^k b = \min(a + b, k)$ saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{\varphi_{\boldsymbol{S}} \in \Phi}^{k} \varphi_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

- What/where is the minimum of $\Phi_{\mathcal{M}}$?

Cost Function Network $\langle \boldsymbol{V}, \Phi, k \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e integer cost functions
- each $\varphi_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by k (finite or infinite)

Joint cost function using $a +^k b = \min(a + b, k)$ saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{\varphi_{\boldsymbol{S}} \in \Phi}^{k} \varphi_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

- What/where is the minimum of $\Phi_{\mathcal{M}}$?

ANITI INRAE

Cost Function Network $\langle \boldsymbol{V}, \Phi, k \rangle$

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e integer cost functions
- each $\varphi_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by k (finite or infinite)

Joint cost function using $a +^k b = \min(a + b, k)$ saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{\varphi_{\boldsymbol{S}} \in \Phi}^{k} \varphi_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

- What/where is the minimum of $\Phi_{\mathcal{M}}$?

Tables (or tensors) for φ_S

- A multidimensional table with a number for every tuple in D^S

Beyond dense tensors

- Weighted clauses: a cost for a cell in a (sparse) tensor
- Names for specific (useful) functions

Soft difference (3 values)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example

$\textsc{WeightedRegular}_{\mathcal{A},S}$

Tables (or tensors) for φ_S

- A multidimensional table with a number for every tuple in D^S

Beyond dense tensors

- Weighted clauses: a cost for a cell in a (sparse) tensor
- Names for specific (useful) functions

Soft difference (3 values)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example

$\textsc{WeightedRegular}_{\mathcal{A},S}$

Tables (or tensors) for φ_S

- A multidimensional table with a number for every tuple in D^S

Beyond dense tensors

- Weighted clauses: a cost for a cell in a (sparse) tensor
- Names for specific (useful) functions

Soft difference (3 values)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example

$\textsc{WeightedRegular}_{\mathcal{A},S}$

COP uses a suitably constrained variable to represent a cost.

Example: Min-CUT

Graph $G = (\boldsymbol{V}, \boldsymbol{E})$ with edge weight function w

- A Boolean variable X_i per vertex $i \in V$
- A cost function per edge $e = (i, j) \in E : \varphi_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$

A simple graph

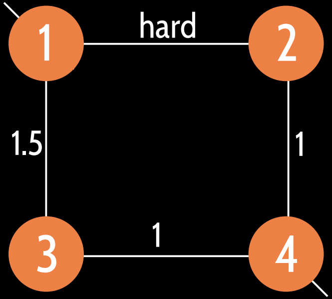
- vertices $\{1, 2, 3, 4\}$
- cut weight 1 or 1.5 $(1, 3)$
- edge $(1, 2)$ hard

Graph $G = (\boldsymbol{V}, \boldsymbol{E})$ with edge weight function w

- A Boolean variable X_i per vertex $i \in V$
- A cost function per edge $e = (i, j) \in E$: $\varphi_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$

A simple graph

- vertices $\{1, 2, 3, 4\}$
- cut weight 1 or 1.5 $(1, 3)$
- edge $(1, 2)$ hard



Min-CUT on 4 variables

```
import pytoulbar2
myCFN = pytoulbar2.CFN(100,1)   # k/ub, resolution
for i in range(4):
    myCFN.AddVariable("x"+str(i+1),["l", "r"]) # returns an index
myCFN.AddFunction(["x1"],[0,100])
myCFN.AddFunction(["x4"],[100,0])
myCFN.AddFunction(["x1","x3"], [0,1.5,1.5,0])
...
sol = myCFN.Solve()  # returns a triple (sol, cost, _)
```

Definition

- Variables X_{ij} for cell (i, j) has domain $\{1, \cdots, 9\}$
- Set R_i (resp. C_j) contains all variables of row i (resp. column j)
- Set S_i contains all variables in sub-cell i
- There is an All-Different constraint on each of these
- or a clique of pairwise different constraints

Example

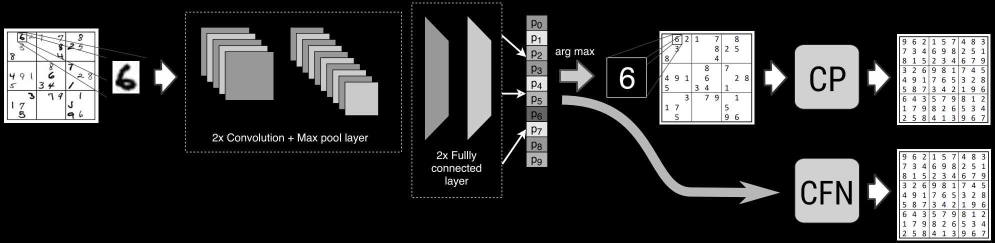
Let's have a look at the pytoulbar2 code.

```
myCFN = pytoulbar2.CFN(1)   # k = 1, so CSP

for i in range(9):
  for j in range(9):
      vIdx = myCFN.AddVariable("X"+str(i+1)+"."+str(j+1),range(1,10))
      columns[j].append(vIdx)
      rows[i].append(vIdx)
      cells[(i//3)*3+(j//3)].append(vIdx)

for scope in rows+columns+cells:
  addCliqueAllDiff(myCFN,scope)   # Adds a clique of pairwise difference

for v,h in enumerate(grid):
  if h:  myCFN.AddFunction([v],[0 if i == h else 1 for i in range(1,10)])
```



The Boolean way

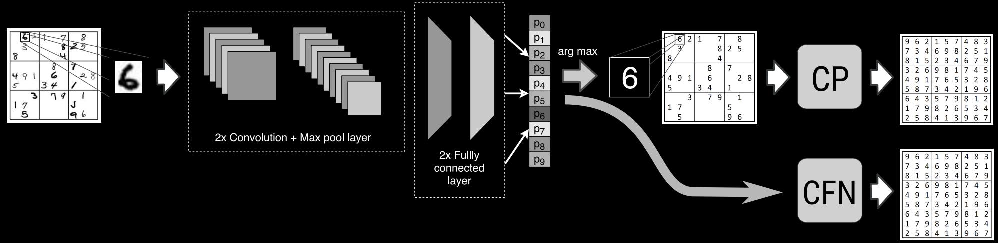
Thanks to Tias Gun for the picture above

1. Assign the cell variable with the prediction
2. LeNet has 99.2% accuracy, SAT-Net dataset 36.2 hints (avg): . . . . . . 74.7% max. accuracy

The Numbers way

1. Add LeNet output logits (negated) as a cost function
2. $(\min \sum -\log) \equiv (\max \prod)$ probabilities . . . . >99% acc.

Numbers: interfacing with ML/DL



The Boolean way

Thanks to Tias Gun for the picture above

1. Assign the cell variable with the prediction
2. LeNet has 99.2% accuracy, SAT-Net dataset 36.2 hints (avg): . . . . . . 74.7% max. accuracy

The Numbers way

1. Add LeNet output logits (negated) as a cost function
2. $(\min \sum - \log) \equiv (\max \prod)$ probabilities . . . . >99% acc.

```
myCFN = pytoulbar2.CFN(1000000000,6)

for i in range(9):
  for j in range(9):
      vIdx = myCFN.AddVariable("X"+str(i+1)+"."+str(j+1),range(1,10))
      columns[j].append(vIdx)
      rows[i].append(vIdx)
      cells[(i//3)*3+(j//3)].append(vIdx)

for scope in rows+columns+cells:
  addCliqueAllDiff(myCFN,scope)   # Adds a clique of pairwise difference

for v, h in enumerate(grid):
    if h: myCFN.AddFunction([v],-MNIST_output(csol,v,h))
```

CFN compared to a COP approach[1,2]

- COP (OR-Tools) + global All-Different
- CFN (toulbar2) + pairwise differences

[1] Maxime Mulamba et al. "**Hybrid Classification and Reasoning for Image-based Constraint Solving**". In: *Proc. of CPAIOR'20, also in arXiv preprint arXiv:2003.11001*. 2020, pp. 364–380.

[2] Céline Brouard, Simon de Givry, and Thomas Schiex. "**Pushing data into CP models using Graphical Model Learning and Solving**". In: *Principles and Practice of Constraint Programming–CP 2020*. Springer, 2020.

CFN compared to a COP approach[1,2]

- COP (OR-Tools) + global All-Different . . . . 0.79"
- CFN (toulbar2) + pairwise differences . . . . 0.05"

[1] Maxime Mulamba et al. "**Hybrid Classification and Reasoning for Image-based Constraint Solving**". In: *Proc. of CPAIOR'20, also in arXiv preprint arXiv:2003.11001*. 2020, pp. 364–380.

[2] Céline Brouard, Simon de Givry, and Thomas Schiex. "**Pushing data into CP models using Graphical Model Learning and Solving**". In: *Principles and Practice of Constraint Programming–CP 2020*. Springer, 2020.

CFN compared to a COP approach[1][2]

- COP (OR-Tools) + global All-Different . . . . 0.79"
- CFN (toulbar2) + pairwise differences . . . . 0.05"
- 99.6% of all problems are solved backtrack-free by toulbar2

[1] Maxime Mulamba et al. "**Hybrid Classification and Reasoning for Image-based Constraint Solving**". In: *Proc. of CPAIOR'20, also in arXiv preprint arXiv:2003.11001*. 2020, pp. 364–380.

[2] Céline Brouard, Simon de Givry, and Thomas Schiex. "**Pushing data into CP models using Graphical Model Learning and Solving**". In: *Principles and Practice of Constraint Programming–CP 2020*. Springer, 2020.

CFN compared to a COP approach[1,2]

- COP (OR-Tools) + global All-Different . . . . 0.79"
- CFN (toulbar2) + pairwise differences . . . . .0.05"
- 99.6% of all problems are solved backtrack-free by toulbar2
- CFN bounds way tighter than COP bounds [25]

[1] Maxime Mulamba et al. **"Hybrid Classification and Reasoning for Image-based Constraint Solving"**. In: *Proc. of CPAIOR'20, also in arXiv preprint arXiv:2003.11001*. 2020, pp. 364–380.

[2] Céline Brouard, Simon de Givry, and Thomas Schiex. **"Pushing data into CP models using Graphical Model Learning and Solving"**. In: *Principles and Practice of Constraint Programming–CP 2020*. Springer, 2020.

Combination and normalisation

- log-linear models (or multiplication)

$$P(t) \propto \exp(-\Phi(t))$$

- optimisation and counting (marginals)
- Normalization is #P-hard.
- High probabilities for good solutions

Markov Random Fields, Bayesian nets, Factor Graphs, Ising/Potts models [19]

- log-linear MRFs $\approx$ CFNs (cost = energy). Image/signal processing.
- Bayes-nets use a DAG product-rule decomposition ($P(A, B) = P(A|B).P(B)$)
- Higher-order factors $\approx$ global constraints

Many theorems and algorithms assume $P(t) > 0$.

Pairwise models [1]

Minimize $\sum_{i,a} \varphi_i(a) \cdot x_{ia} + \sum_{\substack{\varphi_{ij} \in \Phi \\ a \in D^i, b \in D^j}} \varphi_{ij}(a,b) \cdot x_{ia} x_{jb}$ such that

$$\sum_{a \in D^i} x_{ia} = 1 \qquad \forall i \in \{1, \ldots, n\}$$

$$x_{ia} \in \{0,1\} \qquad \forall i \in \{1, \ldots, n\}$$

nd variables, n constraints: concise but challenging

The universal "local polytope" [34, 22, 39, 38] (without eq. (1))

Minimize $\sum_{i,a} \varphi_i(a) \cdot x_{ia} + \sum_{\substack{\varphi_{ij} \in \Phi \\ a \in D^i, b \in D^j}} \varphi_{ij}(a,b) \cdot y_{iajb}$ such that

$$\sum_{a \in D^i} x_{ia} = 1 \qquad \forall i \in \{1, \ldots, n\}$$

$$\sum_{b \in D^j} y_{iajb} = x_{ia} \qquad \forall \varphi_{ij} \in \Phi, \forall a \in D^i$$

$$\sum_{a \in D^i} y_{iajb} = x_{jb} \qquad \forall \varphi_{ij} \in \Phi, \forall b \in D^j$$

$$x_{ia} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\} \quad (1)$$

$nd + ed^2$ variables, $n + 2ed$ constraints: strong but expensive

As an NP-hard discrete optimization framework

Discrete Graphical Models

- Are polynomials in their decision variables (OHE)
- Criterion linear in the finite parameters (costs)
- Criterion and constraints are homogeneous, but
- Additive $(a + a = 2a)$ and idempotent $(a + a = a)$ respectively

Non serial Dynamic Programming [3, 2, 4]

- Tree-decompositions [33] and compilation: OBDD, d-DNNF, SDD, ADD [17] vs. Lauritzen/Shafer hypergraph message passing [35, 24], ...
- Exact, operator flexible (semi-ring), differentiable, but space and time exponential

Local reasoning

- (soft) local consistencies/UP/max-resolution [31, 23] vs. (convergent) belief propagation [39, 21] (LP dual)
- Solves tree-structured/(submodular) min-max closed problems
- Efficient, bounds (convergent), heuristics (CP-BP [13]).

Optimisation/decoding and counting/inference

Tree search

- Backtrack, Branch and Bound, …(optimisation, counting), hashing [14]
- Exact, exponential time.

Random search

- Local search (optimisation) vs. MCMC sampling (counting)
- Asymptotic properties, arbitrary time and quality

Relaxations

- LP [39] and convex SDP [38, 18] (Max2SAT, Pairwise additive GMs)
- Poly. time, differentiable, but approximate.

Deterministic models

From a set of (optimal) solutions (or non-solutions)

- Constraint acquisition [7]
- Preference elicitation [11]
- related to classical inverse optimisation [15] (LP, convex, conic, ILP)

Stochastic models

From a fair sample

- Maximum log-likelihood (Z, CRF loss)
- Structured Perceptron/Hinge losses,...
- Fenchel-Young losses [9, 27, 5, 29] (regularization)
- Mean field/variational [38]
- Approximate [37, 28], Pseudo likelihoods [6]

Thank you all for your attention!

And to all CFN/toulbar2 contributors

S. de Givry (INRAE)	G. Katsirelos (INRAE)	M. Zytnicki (PhD, INRAE)
D. Allouche (INRAE)	M. Ruffini (PhD)	H. Nguyen (PhD)
M. Cooper (IRIT, Toulouse)	J. Larrosa (UPC, Spain)	F. Heras (UPC, Spain)
M. Sanchez (PostDoc)	E. Rollon (UPC, Spain)	P. Meseguer (CSIC, Spain)
G. Verfaillie (ONERA, ret.)	JH. Lee (CU. Hong Kong)	C. Bessiere (LIMM, Montpellier)
JP. Métivier (GREYC, Caen)	S. Loudni (GREYC, Caen)	M. Fontaine (GREYC, Caen)
D. Simoncini (PostDoc, UT1)	C. Viricel (PhD)	C. Terrioux (LSIS)
P. Jégou (LSIS)	A. Ouali (GREYC)	Y. Lebbah (GREYC)
L. Loukil (GREYC)	P. Boizumault (GREYC)	Mario (CU. Hong-Kong)
M. Lemaître (CERT)	L. Lobjois (CERT)	B. Hurley (Insight)
B. Neveu (INRIA, Sophia)	G. Trombettoni (INRIA)	S. Buchet (INRAE)…

Questions?

[1] David Allouche et al. "**Computational protein design as an optimization problem**". In: *Artificial Intelligence* 212 (2014), pp. 59–79.

[2] Umberto Bertele and Francesco Brioschi. "**A new algorithm for the solution of the secondary optimization problem in non-serial dynamic programming**". In: *Journal of Mathematical Analysis and Applications* 27.3 (1969), pp. 565–574.

[3] Umberto Bertele and Francesco Brioschi. "**Contribution to nonserial dynamic programming**". In: *Journal of Mathematical Analysis and Applications* 28.2 (1969), pp. 313–325.

[4] Umberto Bertelé and Francesco Brioshi. ***Nonserial Dynamic Programming***. Academic Press, 1972.

[5] Quentin Berthet et al. "**Learning with Differentiable Perturbed Optimizers**". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9508–9519. url: https://proceedings.neurips.cc/paper/2020/file/6bb56208f672af0dd65451f869fedfd9-Paper.pdf.

[6] Julian Besag. "**Statistical analysis of non-lattice data**". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 24.3 (1975), pp. 179–195.

[7] Christian Bessiere, Clément Carbonnel, and Areski Himeur. "**Learning Constraint Networks over Unknown Constraint Languages**". In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. Ed. by Edith Elkind. Main Track.

International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 1876–1883. DOI: 10.24963/ijcai.2023/208. URL: https://doi.org/10.24963/ijcai.2023/208.

[8] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. **"Semiring-based constraint satisfaction and optimization"**. In: *Journal of the ACM (JACM)* 44.2 (1997), pp. 201–236.

[9] Mathieu Blondel, André FT Martins, and Vlad Niculae. **"Learning with Fenchel-Young losses"**. In: *Journal of Machine Learning Research* 21.35 (2020), pp. 1–69.

[10] E. Boros and P. Hammer. **"Pseudo-Boolean Optimization"**. In: *Discrete Appl. Math.* 123 (2002), pp. 155–225.

[11] Ronen I Brafman, Carmel Domshlak, and Solomon Eyal Shimony. **"On graphical modeling of preference and importance"**. In: *Journal of Artificial Intelligence Research* 25 (2006), pp. 389–424.

[12] Céline Brouard, Simon de Givry, and Thomas Schiex. **"Pushing data into CP models using Graphical Model Learning and Solving"**. In: *Principles and Practice of Constraint Programming–CP 2020*. Springer, 2020.

[13] Auguste Burlats and Gilles Pesant. **"Exploiting entropy in constraint programming"**. In: *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer. 2023, pp. 320–335.

[14] Supratik Chakraborty, Kuldeep S Meel, and Moshe Y Vardi. **"A scalable approximate model counter"**. In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2013, pp. 200–216.

[15] Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. **"Inverse optimization: Theory and applications"**. In: *Operations Research* 73.2 (2023).

[16] Martin Cooper, Simon de Givry, and Thomas Schiex. **"Graphical Models: Queries, Complexity, Algorithms"**. In: *Leibniz International Proceedings in Informatics (STACS'2020)* 154 (2020), pp. 4–1.

[17] Adnan Darwiche and Pierre Marquis. **"Compiling propositional weighted bases"**. In: *Artificial Intelligence* 157.1-2 (2004), pp. 81–113.

[18] Valentin Durante, George Katsirelos, and Thomas Schiex. **"Efficient low rank convex bounds for pairwise discrete Graphical Models"**. In: *Thirty-ninth International Conference on Machine Learning*. Baltimore, United States, July 2022.

[19] Ernst Ising. **"Beitrag zur theorie des ferromagnetismus"**. In: *Zeitschrift für Physik* 31.1 (1925), pp. 253–258.

[20] Daphne Koller and Nir Friedman. ***Probabilistic graphical models: principles and techniques***. MIT press, 2009.

[21] Vladimir Kolmogorov and Carsten Rother. "**Minimizing nonsubmodular functions with graph cuts-a review**". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.7 (2007), pp. 1274–1279.

[22] A M C A. Koster. "**Frequency assignment: Models and Algorithms**". Available at www.zib.de/koster/thesis.html. PhD thesis. The Netherlands: University of Maastricht, Nov. 1999.

[23] Javier Larrosa and Federico Heras. "**Resolution in Max-SAT and its relation to local consistency in weighted CSPs**". In: *IJCAI*. 2005, pp. 193–198. url: http://www.ijcai.org/papers/0360.pdf.

[24] Steffen L Lauritzen and Finn Verner Jensen. "**Local computation with valuations from a commutative semigroup**". In: *Annals of Mathematics and Artificial Intelligence* 21 (1997), pp. 51–69.

[25] Jimmy Ho-Man Lee and Ka Lun Leung. "**Consistency techniques for flow-based projection-safe global cost functions in weighted constraint satisfaction**". In: *Journal of Artificial Intelligence Research* 43.1 (2012), pp. 257–292.

[26] Maxime Mulamba et al. "**Hybrid Classification and Reasoning for Image-based Constraint Solving**". In: *Proc. of CPAIOR'20, also in arXiv preprint arXiv:2003.11001*. 2020, pp. 364–380.

[27] Mathias Niepert, Pasquale Minervini, and Luca Franceschi. "**Implicit MLE: Backpropagating Through Discrete Exponential Family Distributions**". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 14567–14579. url: https://proceedings.neurips.cc/paper/2021/file/7a430339c10c642c4b2251756fd1b484-Paper.pdf.

[28] Youngsuk Park et al. "**Learning the network structure of heterogeneous data via pairwise exponential Markov random fields**". In: *Proceedings of machine learning research* 54 (2017), p. 1302.

[29] Michael E Sander et al. "**Joint learning of energy-based models and their partition function**". In: *arXiv preprint arXiv:2501.18528* (2025).

[30] Bogdan Savchynskyy et al. "**Discrete graphical models—an optimization perspective**". In: *Foundations and Trends® in Computer Graphics and Vision* 11.3-4 (2019), pp. 160–429.

[31] T. Schiex. "**Arc consistency for soft constraints**". In: *Principles and Practice of Constraint Programming - CP 2000*. Vol. 1894. LNCS. Singapore, Sept. 2000, pp. 411–424.

[32] T. Schiex, H. Fargier, and G. Verfaillie. "**Valued Constraint Satisfaction Problems: hard and easy problems**". In: *Proc. of the* 14^{th} *IJCAI*. Montréal, Canada, Aug. 1995, pp. 631–637.

[33] Thomas Schiex. ***A note on CSP graph parameters***. Tech. rep. Citeseer, 1999.

[34] M.I. Schlesinger. "**Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (Syntactic analysis of two-dimensional visual signals in noisy conditions)**". In: *Kibernetika* 4 (1976), pp. 113–130.

[35] G. Shafer. ***An Axiomatic Study of Computation in Hypertrees.*** Working paper 232. Lawrence: University of Kansas, School of Business, 1991.

[36] Prakash P Shenoy and Glenn Shafer. "**Axioms for probability and belief-function propagation**". In: *Machine intelligence and pattern recognition*. Vol. 9. Elsevier, 1990, pp. 169–198.

[37] Martin J Wainwright, Michael I Jordan, et al. "**Graphical models, exponential families, and variational inference**". In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[38] Martin J Wainwright, Michael I Jordan, et al. "**Graphical models, exponential families, and variational inference**". In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[39] T. Werner. "**A Linear Programming Approach to Max-sum Problem: A Review.**". In: *IEEE Trans. on Pattern Recognition and Machine Intelligence* 29.7 (July 2007), pp. 1165–1179. URL: `http://dx.doi.org/10.1109/TPAMI.2007.1036`.