Weighted CP & related framemorks

Soft local consistencies as bounds

Dagsthul 2025

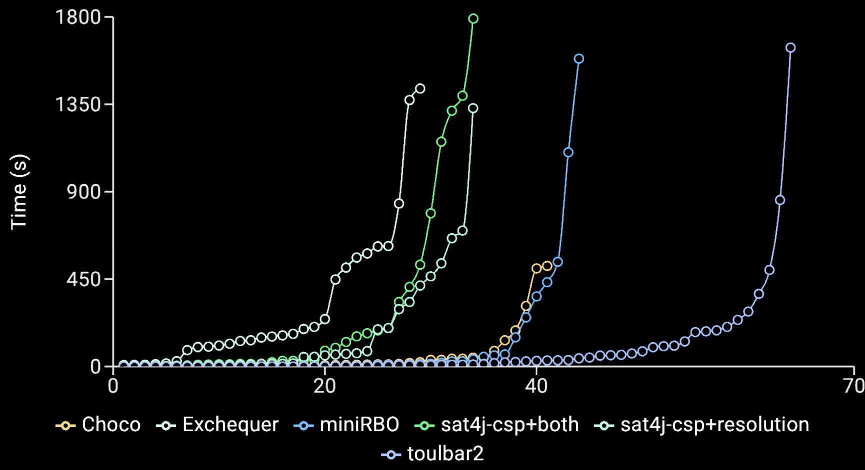
T. Schiex

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

May 5, 2025

XCSP3 MiniCOP track

Problems solved with a proof of optimality



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 $c_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

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

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 $c_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

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

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 $c_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

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

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 $c_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \rightarrow \{t, f\}$

Joint truth function

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

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 $c_{\boldsymbol{S}} \in \Phi$ is a truth function from $D^{\boldsymbol{S}} \to \{t, f\}$

Joint truth function

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

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, global constraints,...

Tables (or tensors) for $c_{\boldsymbol{S}}$

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

Pairwise difference (3 values)

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

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, global constraints,...

Beyond dense tensors

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

Languages for domains and constraints

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

Beyond dense tensors

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

Most famous

AllDifferent*s*

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

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e **non negative** integer cost functions
- each $c_S \in \Phi$ is a numerical function bounded by $\top$ — $c_\varnothing$ lower bound

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

$$\Phi_{\mathcal{M}} = \sum_{c_S \in \Phi}^{\top} c_S$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

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

ANITI INRAE

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

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e **non negative** integer cost functions
- each $c_S \in \Phi$ is a numerical function bounded by $\top$ — $c_\varnothing$ lower bound

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

$$\Phi_{\mathcal{M}} = \sum_{c_S \in \Phi}^{\top} c_S$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

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

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

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e **non negative** integer cost functions
- each $c_S \in \Phi$ is a numerical function bounded by $\top$ — $c_\varnothing$ lower bound

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

$$\Phi_\mathcal{M} = \sum_{c_S \in \Phi}^{\top} c_S$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

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

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

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e **non negative** integer cost functions
- each $c_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by $\top$ — $c_{\varnothing}$ lower bound

Joint cost function using $a +^{\top} b = \min(a + b, \top)$ — saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{c_{\boldsymbol{S}} \in \Phi}^{\top} c_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

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

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

- a sequence of discrete domain variables $\boldsymbol{V}$
- a set Φ of e **non negative** integer cost functions
- each $c_{\boldsymbol{S}} \in \Phi$ is a numerical function bounded by $\top$ — $c_{\varnothing}$ lower bound

Joint cost function using $a +^{\top} b = \min(a + b, \top)$ — saturating arithmetics

$$\Phi_{\mathcal{M}} = \sum_{c_{\boldsymbol{S}} \in \Phi}^{\top} c_{\boldsymbol{S}}$$

The Weighted Constraint Satisfaction Problem (decision NP-complete)

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

Tables (or tensors) for c_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) global cost 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 c_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) global cost functions

Soft difference (3 values)

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

Example

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

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 $e = (i, j) \in E : c_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$

A simple graph

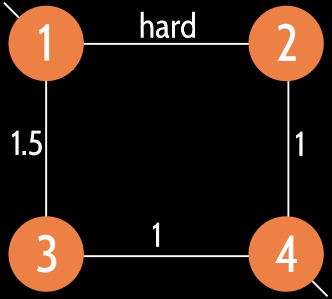
- vertices $\{1, 2, 3, 4\}$
- cut weight 1 or 1.5 $(1, 3)$
- edge $(1, 2)$ hard

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 $e = (i, j) \in E : c_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$

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)   # T/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, _)
```

Expressing probabilities through cost/energy distributions

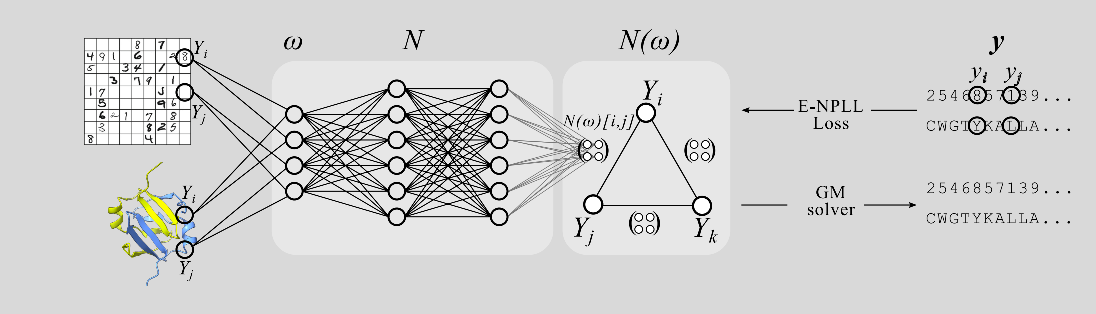
- log-linear models

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

- Normalization is #P-hard
- Optimisation and counting (marginals)

Markov Random Fields, Bayesian nets, Factor Graphs, Ising/Potts models [12]

- log-linear MRFs $\approx$ CFNs (cost = energy). Image/signal processing.
- CFNs can be learned from data.



This can [5, 9, 10]

- Simultaneously learn how to decode numbers and play Sudoku
- Or learn costs from contextual data and solutions (Decision focused learning)
- Or learn how to design new proteins (CoViD neutralizers)

Uses a 0/1 encoding of finite domains: $x_{ia} = 1$ means $x_i = a$.

Pairwise models [2]

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

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

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

nd variables, n constraints: concise but challenging

ANITI INRAE

The "local polytope" [22, 14, 24, 23] (without eq. (1))

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

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

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

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

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

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

A well-studied object

The hope was that local polytopes could be solved faster than LP

Plenty of fast **approximate** Block Coordinate Descent algorithms [20]

Local polytope universality: constructive proof [18]

Any LP can be reduced in linear time to a linear optimization over a local polytope (with infinite weights).

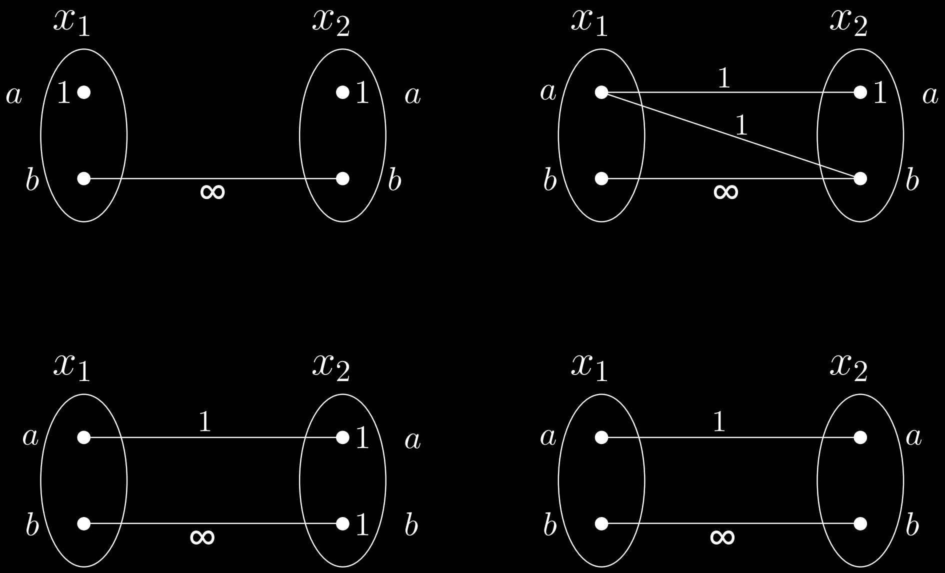
Fundamental for Tree Search efficiency

- Tight bounds
- Efficient bounds

Main difference with CP

- (G)AC: variables and constraints communicate through domains (value deletion)
- Soft (G)AC: variables and cost functions communicate through unary cost functions [21].
- This includes value deletions ($\top$)

Simple example

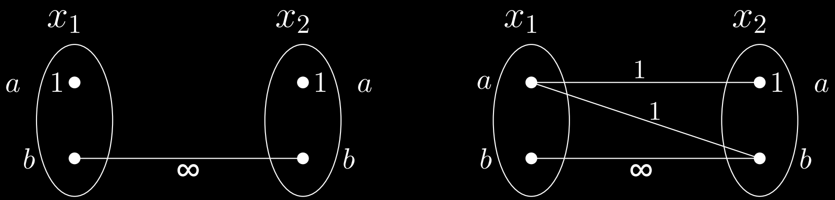


$$c_\varnothing = 1$$

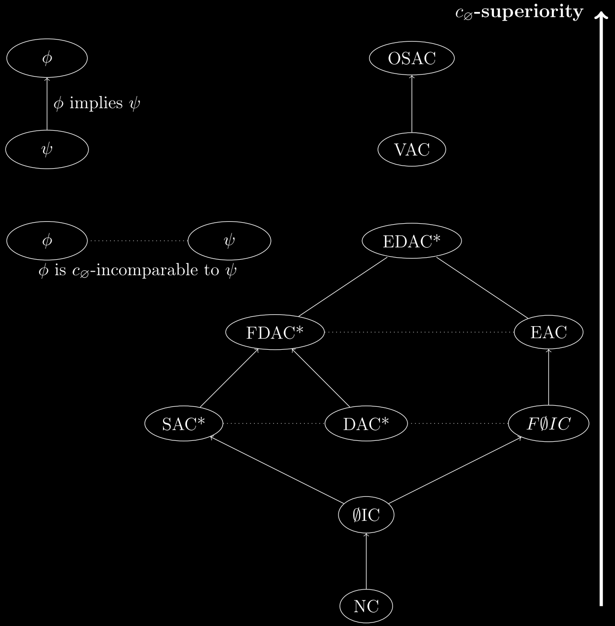
Procedure MoveCost($c_{\boldsymbol{S}}, c_{\boldsymbol{S}'}, \tau, \alpha$): Move α units of cost between the tuple τ of scope $\boldsymbol{S}$ and tuples τ' that extend τ in scope $\boldsymbol{S}'$

Data: Scopes $\boldsymbol{S} \subset \boldsymbol{S}'$
Data: $\tau \in \ell(\boldsymbol{S})$
Data: cost α to move
$c_{\boldsymbol{S}}(\tau) \leftarrow c_{\boldsymbol{S}}(\tau) + \alpha$;
foreach $\tau' \in \ell(\boldsymbol{S}') \mid \tau'_{\boldsymbol{S}} = \tau$ **do**
  $c_{\boldsymbol{S}'}(\tau') \leftarrow c_{\boldsymbol{S}'}(\tau') - \alpha$;

$\mathbf{S} = \{x_1\} \quad \mathbf{S}' = \{x_1, x_2\} \quad \tau = (a) \quad \alpha = -1$



Fix-point algorithms using a well-defined set of Move operations



Algorithm	Time Complexity
NC	$O(nd)$
$\emptyset$IC	$O(ed^2)$
$F\emptyset IC$	$O(ed^2)$
SAC	$O(n^2d^2 + ed^3)$
DAC	$O(ed^2)$
FDAC	$O(end^3)$
EDAC	$O(ed^2max(nd, \top))$
VAC_ε	$O(ed^2\top/\varepsilon)$
OSAC	$O(e^{4.5}d^{5.5}logM)$

Generalize usual (G)AC

- Enforce GAC when applied to constraints
- Most are polytime (some depend on $\top$)
- Produce an equivalent problem (incremental)
- Can have several fixpoints

Virtual Arc Consistency [7]

Definition

A CFN is VAC iff its hardened version has a non empty arc consistent closure.

Hardened: non zero-costs are forbidden ($\top$).

Enforcing VAC — AC-core guided

- Iteratively enforces AC on the hardened version to build a sequence of moves that increase $c_\varnothing$.
- Generalization of Ford Fulkerson (Augmenting DAGs [15])
- With capacity scaling (MaxSAT stratification)
- Solves submodular problems (Min-CUT)

What "Move" operations are used in enforcing Soft AC

- Moving to the 0-ary scope from a unary scope y_i
- Moving between a value a's cost and associated pairs in c_{ij} y_{iaj}
- We should not create negative costs (lower bound)

$$\max \left(c_{\varnothing} + \sum_{i \in \{1,\ldots,n\}} y_i \right)$$

such that:

$$c_i(a) - y_i + \sum_{j \in \{1,\ldots,n\}} y_{iaj} \geq 0 \qquad \forall i \in \{1, \ldots, n\}, \forall a \in D_i$$

$$c_{ij}(ab) - y_{iaj} - y_{jbi} \geq 0 \qquad \forall (i,j), i > j, \forall (a,b) \in D_i \times D_j$$

OSAC is the dual of the local polytope

Soft Arc Consistencies are

- Fast combinatorial heuristics that find good feasible dual solutions
- The reduced costs are used in the reformulated problem
- The problem remains equivalent

VAC vs. LP on Protein design problems

CPLEX V12.4.0.0

```
Problem '3e4h.LP' read.
Root relaxation solution time =  811.28 sec.
...
MIP - Integer optimal solution:  Objective =  150023297067
Solution time =  864.39 sec.
```

tb2 and VAC (AC3 based)

```
loading CFN file: 3e4h.wcsp
Lb after VAC: 150023297067
Preprocessing time: 9.13 seconds.
Optimum: 150023297067 in 129 backtracks, 129 nodes and 9.38 seconds.
```

Could this be useful for ILP?

By reversing the universal polytope construction somehow?

Let P be an LP $\{\min c^T z | Az = b, z \geq 0\}$

$\mathbf{y}$ *a dual solution of P with cost* $\mathbf{y}^T b$

associated reduced costs $rc^{\mathbf{y}} = c - A^T \mathbf{y}$.

. *Then the LP* $\{\min rc^{\mathbf{y}^T} z + \mathbf{y}^T b | Az = b, z \geq 0\}$ *is equivalent to P.*

The ingredients

- An All-different constraint
- Unary cost functions on all involved variables

Reformulate this subproblem to maximally increase the lower bound $c_\varnothing$

How to reformulate?

- Find an optimal dual solution of this Linear Assignment Problem
- Reformulate using the dual solution

The Primal

$$\min \sum_{i,j \in [1,n]} c_i[j] x_{ij} \quad s.t.$$

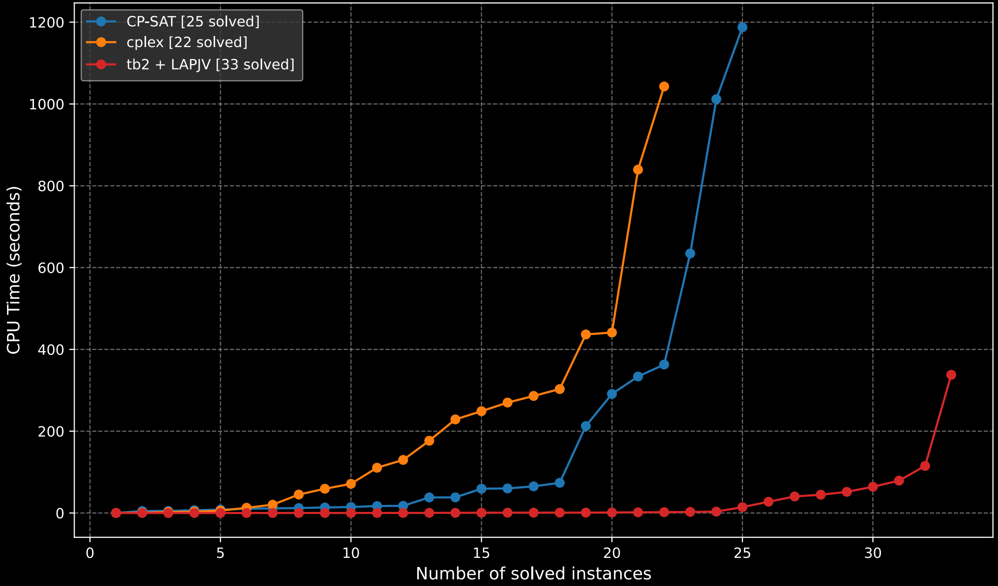
$$\sum_{j \in [1,n]} x_{ij} = 1 \qquad \forall i \in [1,n]$$

$$\sum_{i \in [1,n]} x_{ij} = 1 \qquad \forall j \in [1,n]$$

We can

- solve it using a primal-dual algorithm such as LAPJV (Jonker & Volgenant [13]) algorithm.
- Preserve primal/dual solutions during tree search for incrementality

Quadratic Assignment Problems (QAPLib) $n \leq 100$



We only skimmed through the surface — **Soft Local consistencies**

- On-the-fly variable elimination [16]
- Parallel Hybrid BFS [1, 3]
- Tree decomposition-aware tree-search ÷ [11, 19, 1] (closed several RLFAP instances)
- Dominance analysis [8, 2]
- A parallel complete VNS-LDS-WCP local search algorithm [6, 17]...

Visit toulbar2 websites: `github.com/toulbar2/toulbar2` and `toulbar2.github.io`

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)
P. Montalbano (INRAE)	V. Durante (INRAE)	G. Sewa (INRAE)...

Questions?

[1] David Allouche et al. "**Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP**". In: *Principles and Practice of Constraint Programming*. Springer. 2015, pp. 12–29.

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

[3] Abdelkader Beldjilali et al. "**Parallel hybrid best-first search**". In: *28th International Conference on Principles and Practice of Constraint Programming (CP 2022)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2022, pp. 7–1.

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

[5] Céline Brouard, Simon de Givry, and Thomas Schiex. "**Pushing data into CP models using Graphical Model Learning and Solving**". In: LNCS 4204 (2020).

[6] Antoine Charpentier et al. "**Variable neighborhood search with cost function networks to solve large computational protein design problems**". In: *Journal of chemical information and modeling* 59.1 (2018), pp. 127–136.

[7] M. Cooper et al. "**Soft arc consistency revisited**". In: *Artificial Intelligence* 174 (2010), pp. 449–478.

[8] Simon De Givry, Steven D Prestwich, and Barry O'Sullivan. "**Dead-end elimination for weighted CSP**". In: *Principles and Practice of Constraint Programming*. Springer. 2013, pp. 263–272.

[9] Marianne Defresne, Sophie Barbe, and Thomas Schiex. "**Scalable Coupling of Deep Learning with Logical Reasoning**". 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. 3615–3623. doi: 10.24963/ijcai.2023/402. url: https://doi.org/10.24963/ijcai.2023/402.

[10] Marianne Defresne et al. "**Efficient Neuro-Symbolic Learning of Constraints and Objective**". In: *arXiv preprint arXiv:2508.20978* (2025).

[11] S. de Givry, T. Schiex, and G. Verfaillie. "**Exploiting Tree Decomposition and Soft Local Consistency in Weighted CSP**". In: *Proc. of the National Conference on Artificial Intelligence, AAAI-2006*. 2006, pp. 22–27.

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

[13] R. Jonker and A. Volgenant. "**A shortest augmenting path algorithm for dense and sparse linear assignment problems**". In: *Computing* 38 (1987), pp. 325–340.

[14] 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.

[15] V.K. Koval and M.I. Schlesinger. "Dvumernoe programmirovanie v zadachakh analiza izobrazheniy (Two-dimensional programming in image analysis problems)," in: *USSR Academy of Science, Automatics and Telemechanics* 8 (1976), pp. 149–168.

[16] J. Larrosa. "Boosting search with variable elimination". In: *Principles and Practice of Constraint Programming - CP 2000*. Vol. 1894. LNCS. Singapore, Sept. 2000, pp. 291–305.

[17] Abdelkader Ouali et al. "Variable neighborhood search for graphical model energy minimization". In: *Artificial Intelligence* 278 (2020), p. 103194.

[18] Daniel Prusa and Tomas Werner. "Universality of the local marginal polytope". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 37.4 (2015), pp. 898–904.

[19] M. Sanchez et al. "Russian Doll Search with Tree Decomposition". In: *Proc. IJCAI'09.* San Diego (CA), USA, 2009, pp. 603–608.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.