Optimization in Graphical Models

Connecting NP-complete frameworks

T. Schiex, MIAT, INRA

Many more co-workers and contributors, see bibliography

November 2015 - ICTAI - Vietri sul Mare

The power of NP-complete problem solving

Powering numerous AI areas (several industry related)

1. Model Based Diagnosis, Planning,...
2. Scheduling, Configuration, Resource Allocation,...
3. NLP, Planning/reasoning under uncertainty (*MDP)...

Powering computer (and other) science areas

1. Verification, Cryptography, Testing
2. Image analysis (stereovision, segmentation...)
3. Statistics, Bioinformatics (structural biology)...

Various exact approaches: (weighted) SAT & CSP, ILP, QP...

What is a Graphical Model ?

Informal

1. A set of discrete variables, each with a domain
2. We want to define a joint function on all those variables
3. We do this by combining small functions involving few variables

What is a Graphical Model ?

Informal

1. A set of discrete variables, each with a domain
2. We want to define a joint function on all those variables
3. We do this by combining small functions involving few variables

Why Graphical ?

1. a vertex per variable, a (hyper)edge per function
2. Or its incidence graph (Factor graphs[28])
3. Allows to describe knowledge on a lot of variables concisely
4. Usually hard to manipulate (NP-hard queries).

Constraint Network

1. Set $X \ni x_i$ of n variables, with finite domain D^i ($|D^i| \leq d$)
2. Set $C \ni c_S : D^S \to \{0, 1\}$ of e constraints
3. c_S has scope $S \subset X$ ($|S| \leq r$)
4. Defines a **factorized** joint constraint over X:

$$\forall t \in D^X, C(t) = \max_{c_S \in C} c_S(t[S])$$

A Constraint Network is a GM

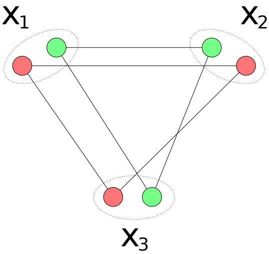
Constraint Network

1. Set $X \ni x_i$ of n variables, with finite domain D^i ($|D^i| \leq d$)
2. Set $C \ni c_S : D^S \to \{0, 1\}$ of e constraints
3. c_S has scope $S \subset X$ ($|S| \leq r$)
4. Defines a **factorized** joint constraint over X:

$$\forall t \in D^X, C(t) = \max_{c_S \in C} c_S(t[S])$$

Graph coloring/RLFAP-feas

1. A graph $G = (V, E)$ and m colors.
2. Can we color all vertices in such a way that no edge connects two vertices of the same color ?



SAT : boolean CN with clauses

SAT

1. Variables x_i are boolean ($0 \equiv \text{true}, 1 \equiv \text{false}$)
2. Constraints are defined as clauses $l_{i_1} \vee \cdots \vee l_{i_p}$
   l_i is x_i or $\bar{x}_i = (1 - x_i)$ (negation).
3. Truth value: $(l_{i_1} \vee \cdots \vee l_{i_p}) = l_{i_1} \times \cdots \times l_{i_p}$

$$\forall t \in D^X, C(t) = \max_{c_S \in C} c_S(t[S])$$

Simply

Shift from boolean functions to cost functions

Lifting CSP to optimization

Cost Function Networks - Weighted Constraint Networks

- Variables and domains as usual
- Cost functions $W \ni c_S : D^S \rightarrow \{0, \ldots, k\}$ (k finite or not)
- Cost combined by (bounded) addition[8] (other, see VCSP[46]).

$$cost(t) = \sum_{c_S \in C} c_S(t[S]) \qquad c_\varnothing : \text{lower bound}$$

A solution has cost $< k$. Optimal if minimum cost.

Lifting CSP to optimization

Cost Function Networks - Weighted Constraint Networks

- Variables and domains as usual
- Cost functions $W \ni c_S : D^S \rightarrow \{0, \dots, k\}$ (k finite or not)
- Cost combined by (bounded) addition[8] (other, see VCSP[46]).

$$cost(t) = \sum_{c_S \in C} c_S(t[S]) \qquad c_\varnothing : \text{lower bound}$$

A solution has cost $< k$. Optimal if minimum cost.

Benefits

- Defines feasibility and cost homogeneously
- A constraint is a cost function with costs in $\{0, k\}$

Weighted Partial Max-SAT

- Just associate a positive integer cost v_i to each clause.
- Cost value: $(l_{i_1} \vee \cdots \vee l_{i_p}, v_i) = v_i \times (l_{i_1} \times \cdots \times l_{i_p})$
- Minimize the sum of all these.
- Polynomial Pseudo-Boolean Optimization.[5]
- A clause with cost k is hard.

Markov Random Fields, Bayesian Networks

- Random variables X with discrete domains
- joint probability distribution $p(X)$ defined through the product of positive real-valued functions:

$$p(X = t) \propto \prod_{c_S \in C} c_S(t[S])$$

Massively used in 2/3D Image Analysis, Statistical Physics, NLP, planning/reasoning under uncertainty...

Markov Random Fields, Bayesian Networks

- Random variables X with discrete domains
- joint probability distribution $p(X)$ defined through the product of positive real-valued functions:

$$p(X = t) \propto \prod_{c_S \in C} c_S(t[S])$$

Massively used in 2/3D Image Analysis, Statistical Physics, NLP, planning/reasoning under uncertainty. . .

Maximum a Posteriori MAP-MRF

MRF$\equiv$ CFN up to a $(-\log)$ transform.

Binary MRF/CFN as 01LP (infinite k, finite costs)

01 LP Variables, for a binary MRF/CFN

1. x_{ia}: value a used for variable x_i.
2. y_{iajb}: pair (a, b) used for x_i and x_j

$$\text{Minimize} \sum_{i,a} c_i(a) \cdot x_{ia} + \sum_{\substack{c_{ij} \in C \\ a \in D^i, b \in D^j}} c_{ij}(a,b) \cdot y_{iajb} \quad \text{subject to}$$

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

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

$$x_{ia}, y_{iajb} \in \{0, 1\}$$

Linear relaxation : the local polytope[47,25,53]

Exact approaches (beyond ILP)

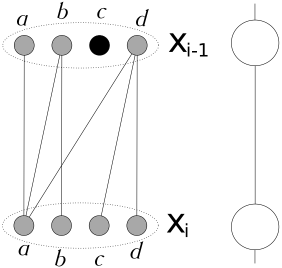
1. Full dynamic programming (Variable elimination,[3,13] resolution[12,43,14])
2. Tree search + fast, incremental approximate local reasoning

Fast, approximate reasoning with some guarantees

- Arc Consistency and Unit Propagation (CSP/SAT)
- Message Passing (MRF/BN)
- Soft Arc Consistency in CFN and UP in PWMaxSAT

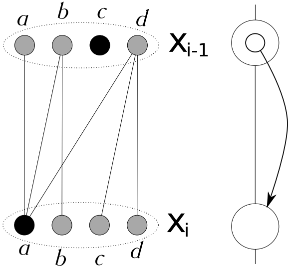
AC as Dynamic programming

- Imagine a CSP with linear graph
- Which values of x_i belong to a solution of $x_1, \ldots x_i$ knowing those for x_{i-1}.
- $\min_{x_{i-1}}(\max(c_{i-1}, c_{i,i-1}))$



AC as Dynamic programming

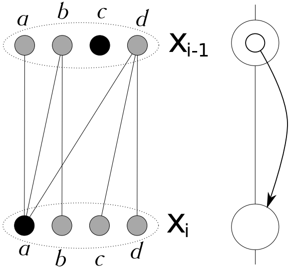
- Imagine a CSP with linear graph
- Which values of x_i belong to a solution of $x_1, \ldots x_i$ knowing those for x_{i-1}.
- $\min_{x_{i-1}}(\max(c_{i-1}, c_{i,i-1}))$



Arc Consistency = local dynamic programming

AC as Dynamic programming

- Imagine a CSP with linear graph
- Which values of x_i belong to a solution of $x_1, \ldots x_i$ knowing those for x_{i-1}.
- $\min_{x_{i-1}}(\max(c_{i-1}, c_{i,i-1}))$



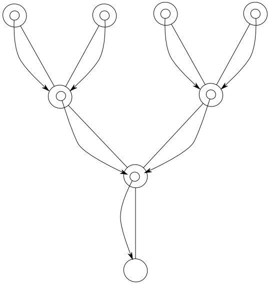
Revise = Equivalence Preserving Transformation (EPT)

- $\nexists b \in D^j \mid c_{ij}(a, b) = 0$.
- we can delete a in c_i (or D^i).
- the resulting problem is equivalent (same set of solutions)

(Directional) AC solves Berge-acyclic CN

Rooted tree CN

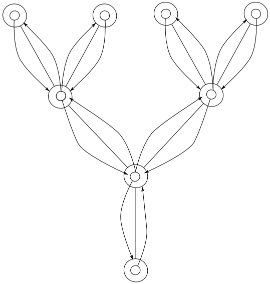
- Revise from leaves to root
- Root domain: values that belong to a solution



(Directional) AC solves Berge-acyclic CN

Tree CN

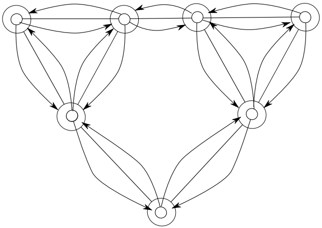
- Revise from leaves and back
- All domains: values that belong to a solution
- Resulting problem solved backtrack-free[20,19]



Can be done on any CN, with arbitrary graph

Arc consistency (Waltz 72)

1. Linear time (tables)
2. Unique fixpoint (confluent)
3. Preserves equivalence
4. May detect infeasibility
5. Problem transformation (incremental)



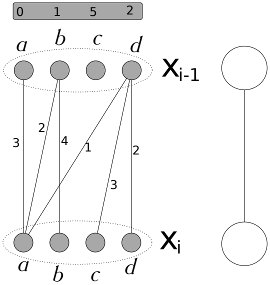
Maintaining AC during search.

1. Same update equation.
2. One litteral and one clause.
3. $\min_{x_{i-1}}(\max(c_{i-1}, c_{i,i-1}))$

Maintaining UP during search (DPLL, ancestor of CDCL)

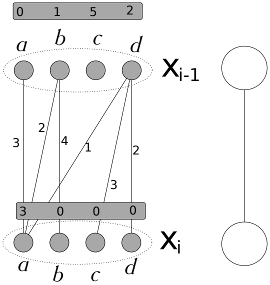
MP - Dynamic Programming[40,28]

1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



MP - Dynamic Programming[40,28]

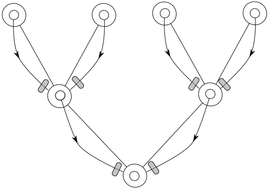
1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



MRF/BN/Factor graphs ($-$ log domain)

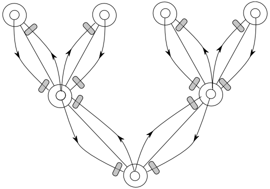
MP - Dynamic Programming[40,28]

1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



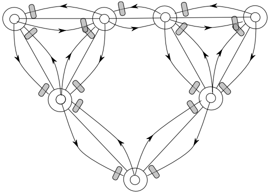
MP - Dynamic Programming[40,28]

1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



MP - Dynamic Programming[40,28]

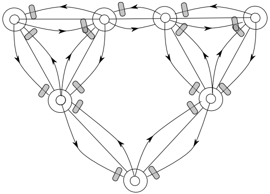
1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



MRF/BN/Factor graphs ($-\log$ domain)

MP - Dynamic Programming[40,28]

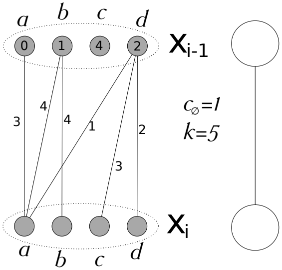
1. optimum cost from x_1 to $a \in D_i$ knowing those for x_{i-1}
2. $\min_{x_{i-1}}(c_{i-1} + c_{i,i-1})$
3. Use external functions (messages) to store DP results



1. Solves Berge acyclic MRF/BN (acyclic Factor Graphs)
2. Does not converge on graphs (Loopy Belief Propagation)
3. Massively used to produce "good" solutions (turbo-decoding[42])
4. More recently introduced in Distributed COP[18]
5. Not an equivalence preserving transformation[40]

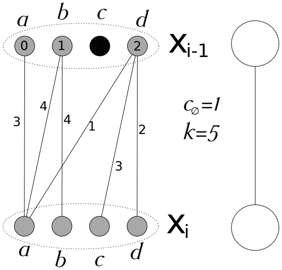
Soft AC as Dynamic programming

1. MRF message passing but...
2. use $c_\varnothing$ and $c_i(\cdot)$ to store optimum cost from x_1 to x_i
3. Preserves equivalence by "cost shifting"[47,52,45]



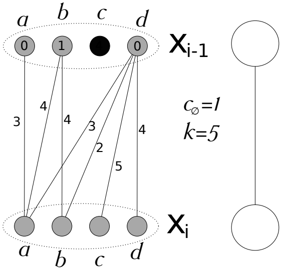
Soft AC as Dynamic programming

1. MRF message passing but...
2. use $c_\varnothing$ and $c_i(\cdot)$ to store optimum cost from x_1 to x_i
3. Preserves equivalence by "cost shifting"[47,52,45]



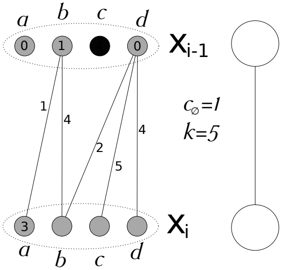
Soft AC as Dynamic programming

1. MRF message passing but...
2. use $c_\varnothing$ and $c_i(\cdot)$ to store optimum cost from x_1 to x_i
3. Preserves equivalence by "cost shifting"[47,52,45]



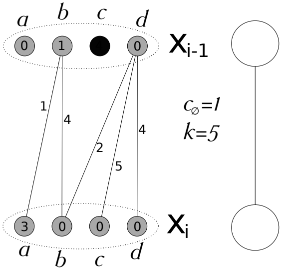
Soft AC as Dynamic programming

1. MRF message passing but. . .
2. use $c_\varnothing$ and $c_i(\cdot)$ to store optimum cost from x_1 to x_i
3. Preserves equivalence by "cost shifting"[47,52,45]



Soft AC as Dynamic programming

1. MRF message passing but...
2. use $c_\varnothing$ and $c_i(\cdot)$ to store optimum cost from x_1 to x_i
3. Preserves equivalence by "cost shifting"[47,52,45]



Equivalence Preserving Transformation

Arc EPT: `Project` $(\{ij\}, \{i\}, a, \alpha)$

- Shifts α units of cost between $c_i(a)$ and c_{ij}.
- Shift direction: sign of α.
- α constrained: no negative costs!

Precondition: $-c_i(a) \leq \alpha \leq \min_{t' \in D^{ij}, t'[i]=a} c_{ij}(t')$;

Procedure `Project`$(\{i, j\}, \{i\}, a, \alpha)$

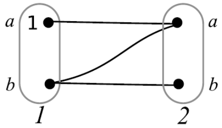
  $c_i(a) \leftarrow c_i(a) + \alpha$;

  foreach *($t' \in D^{ij}$ such that $t'[i] = a, c_{ij}(t') < k$)* **do**

    $c_{ij}(t') \leftarrow c_{ij}(t') - \alpha$;

  end

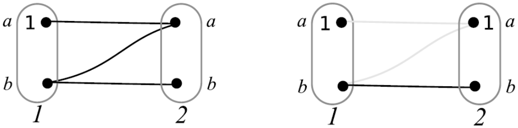
Example



Example

$$\texttt{Project}(\{1,2\},\{2\},a,1)$$

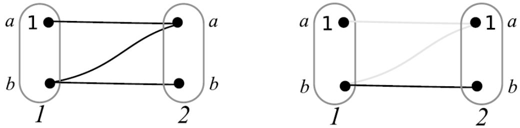
$$\rightarrow$$



Example

$\texttt{Project}(\{1,2\},\{2\},a,1)$

$\rightarrow$

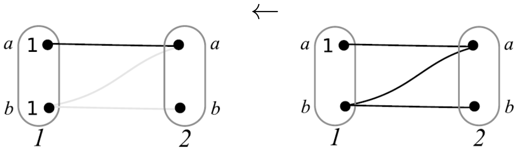


$\leftarrow$

$\texttt{Project}(\{1,2\},\{2\},a,-1)$

Example

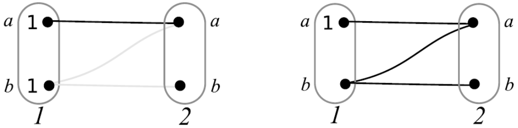
$\texttt{Project}(\{1,2\},\{1\},b,1)$



Example

$\texttt{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$



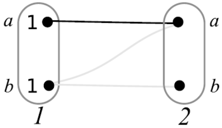
$\rightarrow$

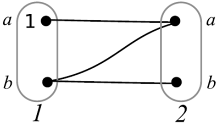
$\texttt{Project}(\{1,2\},\{1\},b,-1)$

Example

$\texttt{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$

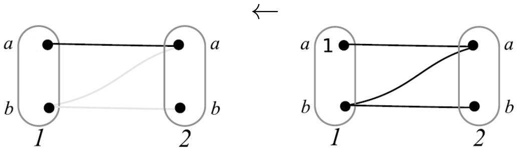




$\Downarrow \quad \texttt{Project}\left(\{1\},\varnothing,[],1\right)$

Example

$\texttt{Project}(\{1,2\},\{1\},b,1)$



$\Downarrow \qquad \texttt{Project}\left(\{1\},\varnothing,[],1\right)$

$c_\varnothing = 1$

Properties

- Solves tree structured problems (ordering), optimum in $c_\varnothing$
- Reformulation: incremental
- May loop indefinitely (graphs)
- No unique fixpoint (when it exists)
- Exploited since the 70s by the Ukrainian school[47,27,26,53] and for a subclass of ILP.[52]

Breaking the loops

1. Arc consistency: prevent loops at the arc level[45]
2. Node consistency[30]
3. Directional AC: prevent loops at a global level[6,32,33]
4. Combine AC and DAC into FDAC[32,33]
5. Pool costs from all stars to $c_\varnothing$ in EAC[34]
6. Combine AC+DAC+EAC in EDAC[34]

All $O(ed)$ space. Equivalent to their "CSP" counterpart on constraints.

Max SAT Unit propagation (resolution)

Semantically, it's the the same

- Problems with leftovers after cost shifting
- Includes non CNF (Heras, Larrosa[31])
- Represented as a linear number of "compensation clauses"[4]

Finding an optimal order[8]

Finding an optimal sequence of integer arc EPTs that maximizes the lower bound is NP-hard.

Finding an optimal order[8]

Finding an optimal sequence of integer arc EPTs that maximizes the lower bound is NP-hard.

Finding an optimal set[7]

Finding an optimal *set* of *rational* arc EPTs that maximizes the lower bound is in P.
This is achieved by solving an LP (OSAC, finite costs, $k = \infty$).

LP Variables, for a binary CFN

1. u_i: amount of cost shifted from c_i to $c_\varnothing$
2. p_{ija}: amount of cost shifted from c_{ij} to $a \in D^i$

Optimal Soft Arc Consistency (finite costs, $k = \infty$)

LP Variables, for a binary CFN

1. u_i: amount of cost shifted from c_i to $c_\varnothing$
2. p_{ija}: amount of cost shifted from c_{ij} to $a \in D^i$

OSAC

Maximize $\sum_{i=1}^{n} u_i$ subject to

$$c_i(a) - u_i + \sum_{(c_{ij} \in C)} p_{ija} \geq 0 \quad \forall i \in \{1, \ldots, n\}, \ \forall a \in D^i$$

$$c_{ij}(a, b) - p_{ija} - p_{jib} \geq 0 \qquad \forall c_{ij} \in C, \forall (a, b) \in D^{ij}$$

See [47, 25, 7, 53, 11].

OSAC is the dual of the local polytope

01 LP Variables, for a binary CFN

1. x_{ia}: value a used for variable x_i.
2. y_{iajb}: pair (a, b) used for x_i and x_j

The MRF local polytope[53]

$$\text{Minimize} \sum_{i,a} c_i(a) \cdot x_{ia} + \sum_{\substack{c_{ij} \in C \\ a \in D^i, b \in D^j}} c_{ij}(a,b) \cdot y_{iajb} \quad \text{s.t}$$

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

$$\sum_{b \in D^j} y_{iajb} - x_{ia} = 0 \qquad \forall c_{ij}/c_{ji} \in C, \forall a \in D^i \qquad (2)$$

u_i multiplier for (1) and p_{ija} for (2).

Better understanding

1. Soft ACs (MP with reformulation) are approximate greedy Block Coordinate Descent solvers of the dual LP (OSAC).
2. They find feasible (but non necessarily optimal) solutions of the dual.
3. optimal does not mean more efficient for tree search.

Generality of the local polytope

2015

Prusa and Werner[41] showed that any "normal" LP can be reduced to such a polytope in linear time (constructive proof).

Could soft arc consistency/MP speed-up LP?

Can we organize our EPTs better w/o LP?

Bool(P)[10]

Given a CFN $P = (X, D, C, k)$
Bool(P) is the CSP $(X, D, C - \{c_\varnothing\}, 1)$.

Bool(P) forbids all positive cost assignments, ignoring $c_\varnothing$.

Can we organize our EPTs better w/o LP?

Bool(P)[10]

Given a CFN $P = (X, D, C, k)$
Bool(P) is the CSP $(X, D, C - \{c_\varnothing\}, 1)$.

Bool(P) forbids all positive cost assignments, ignoring $c_\varnothing$.

Virtual AC

A CFN P is Virtual AC iff Bool(P) has a non empty AC closure.

Can we organize our EPTs better w/o LP?

Bool(P)[10]

Given a CFN $P = (X, D, C, k)$
Bool(P) is the CSP $(X, D, C - \{c_\varnothing\}, 1)$.

Bool(P) forbids all positive cost assignments, ignoring $c_\varnothing$.

Virtual AC

A CFN P is Virtual AC iff Bool(P) has a non empty AC closure.

Virtual AC and MP

TRW-S,[24] MPLP1,[49] SRMP,[23] Max-Sum diffusion,[27,11] Aug-DAG[26]
converge to fixpoints that satisfy the same property.

Properties of VAC

Solutions of Bool(P) are optimal in P.

Properties of VAC

Solutions of Bool(P) are optimal in P.

VAC

1. solves tree-structured problems,
2. solves CFNs with submodular cost functions (Monge matrices)
3. solves CFNs for which AC is a decision procedure in Bool(P).
4. if P is VAC and one value a in each domain such that $c_i(a) = 0$ is solved.
5. There is always at least one such value (or else not VAC).

In practice - Solvers

toulbar2, daoopt, AbsCons (Depth first tree search)
MaxHS, wpm1, wpm2, akmaxsat, minimaxsat. . . (DFS)
ILP-Cplex, QP-Cplex, SDP-BiqMac (Best first tree search)
OpenGM2 (MRF algorithms, Message passing and more).

CELAR 06, $n = 100, d = 44$ - one core

1. 1997: 26 days of a Sun UltraSparc 167 MHz.
2. 2015: optimum found in 7", proved in 73" (2.1GHz CPU)

- 12.5 fold increase in frequency (+architecture)
- More than 30,000 times faster (now easy problem).
- All min-interference CELAR instances closed (see fap.zib.de)

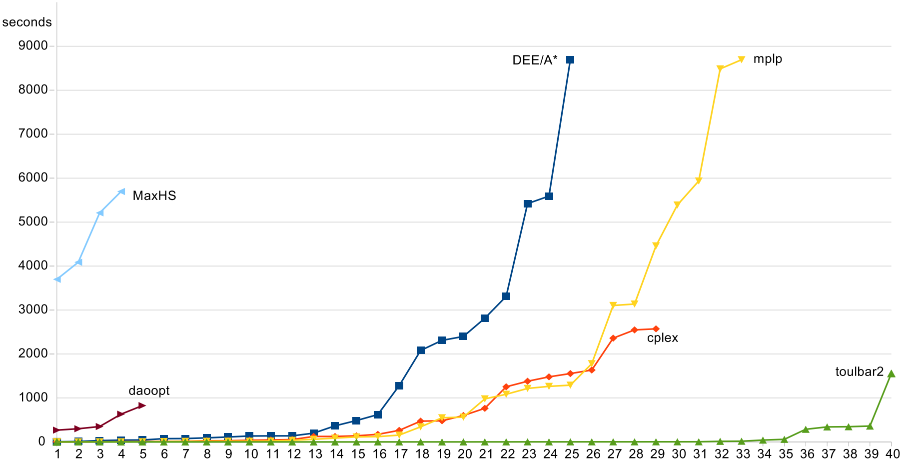
Computational Protein Design[51,1]

Design new enzymes for biofuels, drugs... cosmetics too

- Reduced to a non convex mixed optimization problem
- Discretized (non convexity) leading to a NP-hard...
- binary MAP-MRF capturing molecule stability based on atom-scale forces (electrostatics...)

- Few variables (from 10 to few hundreds)
- Huge domains (typ. $d = 450$)
- Exact solvers: A^*+substituability (DEE[15]), ILP[22]
- By far most used: simulated annealing (Rosetta[21]).

Multi-paradigm comparison - QP,SDP,ILP,WMaxsat,MRF,CFN



Here, VAC faster than LP, often close/same bound

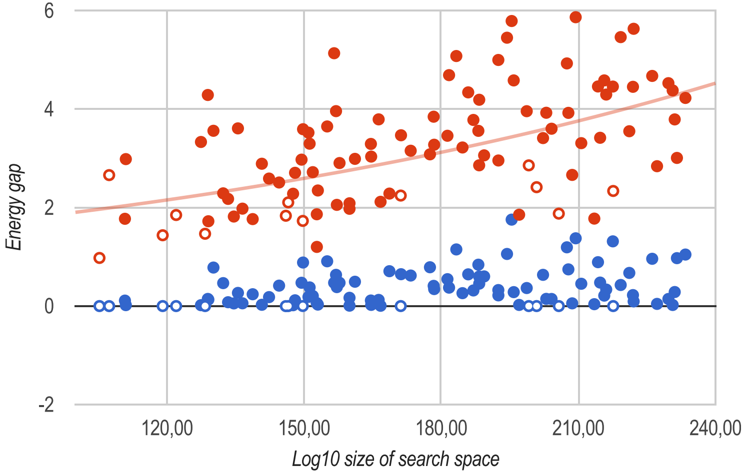
CPLEX V12.4.0.0

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

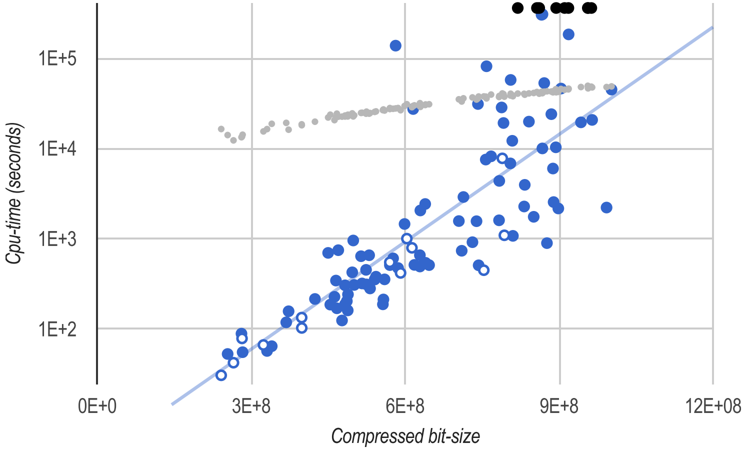
tb2 and VAC

```
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.
```

Exact (simulated annealing is stochastic)[48]



Faster than dedicated simulated annealing[48]



1. First/second in approximate graphical model MRF/MAP challenges (2010, 2012, 2014).
2. Global cost functions (weighted Regular, AllDiff, GCC...)
3. Bioinformatics: pedigree debugging,[44] Haplotyping (QTLMap), structured RNA gene finding[54]
4. Inductive Logic Programming,[2] Natural Langage Processing (in hltdi-l3), Multi-agent and cost-based planning,[29,9] Model Abstraction,[50] diagnostic,[36] Music processing and Markov Logic,[39,38] Data mining,[37] Partially observable Markov Decision Processes,[16] Probabilistic counting[17] and inference,[35] ...

Everything is connected :-)

Questions ?

References I

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

Érick Alphonse and Céline Rouveirol. "Extension of the top-down data-driven strategy to ILP". In: *Inductive Logic Programming*. Springer, 2007, pp. 49–63.

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

Marıa Luisa Bonet, Jordi Levy, and Felip Manyà. "Resolution for max-sat". In: *Artificial Intelligence* 171.8 (2007), pp. 606–618.

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

M C. Cooper. "Reduction operations in fuzzy or valued constraint satisfaction". In: *Fuzzy Sets and Systems* 134.3 (2003), pp. 311–342.

M C. Cooper, S. de Givry, and T. Schiex. "Optimal soft arc consistency". In: *Proc. of IJCAI'2007*. Hyderabad, India, Jan. 2007, pp. 68–73.

M C. Cooper and T. Schiex. "Arc consistency for soft constraints". In: *Artificial Intelligence* 154.1-2 (2004), pp. 199–227.

- Martin C Cooper, Marie de Roquemaurel, and Pierre Régnier. "A weighted CSP approach to cost-optimal planning". In: *Ai Communications* 24.1 (2011), pp. 1–29.
- Martin C Cooper et al. "Virtual Arc Consistency for Weighted CSP." In: *AAAI*. Vol. 8. 2008, pp. 253–258.
- M. Cooper et al. "Soft arc consistency revisited". In: *Artificial Intelligence* 174 (2010), pp. 449–478.
- Martin Davis and Hilary Putnam. "A computing procedure for quantification theory". In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 201–215.
- Rina Dechter. "Bucket Elimination: A Unifying Framework for Reasoning". In: *Artificial Intelligence* 113.1–2 (1999), pp. 41–85.
- Rina Dechter and Irina Rish. "Directional resolution: The Davis-Putnam procedure, revisited". In: *KR* 94 (1994), pp. 134–145.
- J Desmet et al. "The dead-end elimination theorem and its use in protein side-chain positioning." In: *Nature* 356.6369 (Apr. 1992), pp. 539–42. ISSN: 0028-0836. URL: http://www.ncbi.nlm.nih.gov/pubmed/21488406.

Jilles Steeve Dibangoye et al. “Optimally solving Dec-POMDPs as continuous-state MDPs”. In: *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press. 2013, pp. 90–96.

Stefano Ermon et al. “Embed and project: Discrete sampling with universal hashing”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 2085–2093.

Alessandro Farinelli et al. “Decentralised coordination of low-power embedded devices using the max-sum algorithm”. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems. 2008, pp. 639–646.

Eugene C. Freuder. “A sufficient Condition for Backtrack-Bounded Search”. In: *Journal of the ACM* 32.14 (1985), pp. 755–761.

Eugene C. Freuder. “A sufficient Condition for Backtrack-free Search”. In: *Journal of the ACM* 29.1 (1982), pp. 24–32.

Dominik Gront et al. “Generalized fragment picking in Rosetta: design, protocols and applications”. In: *PloS one* 6.8 (2011), e23294.

References IV

Carleton L Kingsford, Bernard Chazelle, and Mona Singh. "Solving and analyzing side-chain positioning problems using linear and integer programming." In: *Bioinformatics (Oxford, England)* 21.7 (Apr. 2005), pp. 1028–36. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti144. URL: http://www.ncbi.nlm.nih.gov/pubmed/15546935.

Vladimir Kolmogorov. "A new look at reweighted message passing". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 37.5 (2015), pp. 919–930.

Vladimir Kolmogorov. "Convergent tree-reweighted message passing for energy minimization". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.10 (2006), pp. 1568–1583.

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.

VK Koval' and Mykhailo Ivanovich Schlesinger. "Two-dimensional programming in image analysis problems". In: *Avtomatika i Telemekhanika* 8 (1976), pp. 149–168.

VA Kovalevsky and VK Koval. "A diffusion algorithm for decreasing energy of max-sum labeling problem". In: *Glushkov Institute of Cybernetics, Kiev, USSR* (1975).

References V

Frank R Kschischang, Brendan J Frey, and Hans-Andrea Loeliger. "Factor graphs and the sum-product algorithm". In: *Information Theory, IEEE Transactions on* 47.2 (2001), pp. 498–519.

Akshat Kumar and Shlomo Zilberstein. "Point-based backup for decentralized POMDPs: Complexity and new algorithms". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2010, pp. 1315–1322.

J. Larrosa. "On Arc and Node Consistency in weighted CSP". In: *Proc. AAAI'02*. Edmondton, (CA), 2002, pp. 48–53.

J. Larrosa and F. Heras. "Resolution in Max-SAT and its relation to local consistency in weighted CSPs". In: *Proc. of the 19th IJCAI*. Edinburgh, Scotland, 2005, pp. 193–198.

J. Larrosa and T. Schiex. "In the quest of the best form of local consistency for Weighted CSP". In: *Proc. of the 18th IJCAI*. Acapulco, Mexico, Aug. 2003, pp. 239–244.

Javier Larrosa and Thomas Schiex. "Solving weighted CSP by maintaining arc consistency". In: *Artif. Intell.* 159.1-2 (2004), pp. 1–26.

J. Larrosa et al. "Existential arc consistency: getting closer to full arc consistency in weighted CSPs". In: *Proc. of the 19th IJCAI*. Edinburgh, Scotland, Aug. 2005, pp. 84–89.

Paul Maier, Dominik Jain, and Martin Sachenbacher. "Compiling AI engineering models for probabilistic inference". In: *KI 2011: Advances in Artificial Intelligence*. Springer, 2011, pp. 191–203.

Paul Maier, Dominik Jain, and Martin Sachenbacher. "Diagnostic hypothesis enumeration vs. probabilistic inference for hierarchical automata models". In: *the International Workshop on Principles of Diagnosis (DX), Murnau, Germany*. 2011.

Jean-Philippe Métivier, Samir Loudni, and Thierry Charnois. "A constraint programming approach for mining sequential patterns in a sequence database". In: *Proceedings of the ECML/PKDD Workshop on Languages for Data Mining and Machine Learning*. arXiv preprint arXiv:1311.6907. Praha, Czech republic, 2013.

Hélene Papadopoulos and George Tzanetakis. "Exploiting structural relationships in audio music signals using Markov Logic Networks". In: *ICASSP 2013-38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Canada (2013)*. 2013, pp. 4493–4497.

Hélène Papadopoulos and George Tzanetakis. “Modeling Chord and Key Structure with Markov Logic.” In: *Proc. Int. Conf. of the Society for Music Information Retrieval (ISMIR)*. 2012, pp. 121–126.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems, Networks of Plausible Inference*. Palo Alto: Morgan Kaufmann, 1988.

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.

Thomas J Richardson and Rüdiger L Urbanke. “The capacity of low-density parity-check codes under message-passing decoding”. In: *Information Theory, IEEE Transactions on* 47.2 (2001), pp. 599–618.

J. Alan Robinson. “A machine-oriented logic based on the resolution principle”. In: *Journal of the ACM* 12 (1965), pp. 23–44.

Martı Sánchez, Simon de Givry, and Thomas Schiex. “Mendelian Error Detection in Complex Pedigrees Using Weighted Constraint Satisfaction Techniques”. In: *Constraints* 13.1-2 (2008), pp. 130–154.

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.

References VIII

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

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.

D. Simoncini et al. "Guaranteed Discrete Energy Optimization on Large Protein Design Problems". In: *Journal of Chemical Theory and Computation* (Nov. 2015).

David Sontag et al. "Tightening LP relaxations for MAP using message passing". In: *arXiv preprint arXiv:1206.3288* (2012).

Peter Struss, Alessandro Fraracci, and D Nyga. "An Automated Model Abstraction Operator Implemented in the Multiple Modeling Environment MOM". In: *25th International Workshop on Qualitative Reasoning, Barcelona, Spain*. 2011.

Seydou Traoré et al. "A new framework for computational protein design through cost function network optimization". In: *Bioinformatics* 29.17 (2013), pp. 2129–2136.

References IX

Dag Wedelin. "An algorithm for large scale 0–1 integer programming with application to airline crew scheduling". In: *Annals of operations research* 57.1 (1995), pp. 283–301.

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.

Matthias Zytnicki, Christine Gaspin, and Thomas Schiex. "DARN! A weighted constraint solver for RNA motif localization". In: *Constraints* 13.1-2 (2008), pp. 91–109.