



HAL
open science

Constraint Programming and Graphical models - Pushing data into your models, The protein design case.

Thomas Schiex, Sophie Barbe, David Simoncini, Jelena Vucinic, Manon
Ruffini, David Allouche

► To cite this version:

Thomas Schiex, Sophie Barbe, David Simoncini, Jelena Vucinic, Manon Ruffini, et al.. Constraint Programming and Graphical models - Pushing data into your models, The protein design case.. 23rd International Symposium on Mathematical Programming (ISMP-18), Jul 2018, Bordeaux, France. hal-02154354

HAL Id: hal-02154354

<https://hal.science/hal-02154354v1>

Submitted on 5 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Constraint Programming and Graphical models

Pushing data into your models

The protein design case

T. Schiex, S. Barbe, D. Simoncini, J. Vucinic, M. Ruffini
Presented by D. Allouche

INRA MIAT, Toulouse, France

July 2018

Constraint network (X, C)

Feasibility biased

- a sequence X of discrete variables x_i , domain D_i

Constraint network (X, C)

Feasibility biased

- a sequence X of discrete variables x_i , domain D_i
- a set C of constraints

Constraint network (X, C)

Feasibility biased

- a sequence X of discrete variables x_i , domain D_i
- a set C of constraints
- $c_S \in C$ involves variables in $S \subseteq X$ and is a boolean function $\prod_{i \in S} D_i \rightarrow \{t, f\}$

Constraint network (X, C)

Feasibility biased

- a sequence X of discrete variables x_i , domain D_i
- a set C of constraints
- $c_S \in C$ involves variables in $S \subseteq X$ and is a boolean function $\prod_{i \in S} D_i \rightarrow \{t, f\}$
- a solution is an assignment of X that satisfies all constraints (NP-complete)

Constraint network (X, C)

Feasibility biased

- a sequence X of discrete variables x_i , domain D_i
- a set C of constraints
- $c_S \in C$ involves variables in $S \subseteq X$ and is a boolean function $\prod_{i \in S} D_i \rightarrow \{t, f\}$
- a solution is an assignment of X that satisfies all constraints (NP-complete)

Constraint programming

- Algorithms to find a solution (Backtrack, constraint propagation)
- Predefined constraints (AllDifferent,...)

Cost function network (X, W)

Homogeneous feasibility and criteria

- a sequence X of discrete variables x_i , domain D_i

Cost function network (X, W)

- a sequence X of discrete variables x_i , domain D_i
- a set W of cost functions

Homogeneous feasibility and criteria

Cost function network (X, W)

- a sequence X of discrete variables x_i , domain D_i
- a set W of cost functions
- $w_S \in W$ is a numerical function $\prod_{i \in S} D_i$

Homogeneous feasibility and criteria

(possibly infinite costs)

Cost function network (X, W)

Homogeneous feasibility and criteria

- a sequence X of discrete variables x_i , domain D_i
- a set W of cost functions
- $w_S \in W$ is a numerical function $\prod_{i \in S} D_i$ (possibly infinite costs)
- a solution optimizes the joint cost $W(X) = \sum_{w_S \in W} w_S(X[S])$ (WCSP, NP-complete)

Cost function network (X, W)

Homogeneous feasibility and criteria

- a sequence X of discrete variables x_i , domain D_i
- a set W of cost functions
- $w_S \in W$ is a numerical function $\prod_{i \in S} D_i$ (possibly infinite costs)
- a solution optimizes the joint cost $W(X) = \sum_{w_S \in W} w_S(X[S])$ (WCSP, NP-complete)

Generalizes CP: a constraint is a cost function that maps to $\{0, \infty\}$

Cost function network (X, W)

Homogeneous feasibility and criteria

- a sequence X of discrete variables x_i , domain D_i
- a set W of cost functions
- $w_S \in W$ is a numerical function $\prod_{i \in S} D_i$ (possibly infinite costs)
- a solution optimizes the joint cost $W(X) = \sum_{w_S \in W} w_S(X[S])$ (WCSP, NP-complete)

Generalizes CP: a constraint is a cost function that maps to $\{0, \infty\}$

Solvers: daopt, toulbar2, MaxHS (MaxSAT)...

- Algorithms to find a solution (Branch and bound, cost function propagation)
- Predefined cost functions (Weighted All-Different,...)

Graph $G = (V, E)$ with edge weight function w

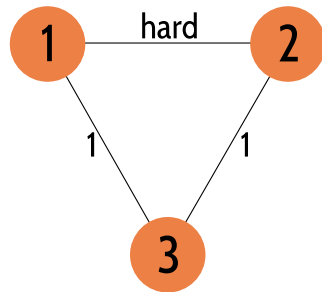
- A boolean variable x_i per vertex $i \in V$
- A cost function per edge $e = (i, j) \in E : w_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$
- Hard edges: constraints with costs 0 or $-\infty$ (when $x_i \neq x_j$)

Graph $G = (V, E)$ with edge weight function w

- A boolean variable x_i per vertex $i \in V$
- A cost function per edge $e = (i, j) \in E : w_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$
- Hard edges: constraints with costs 0 or $-\infty$ (when $x_i \neq x_j$)

3-clique

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

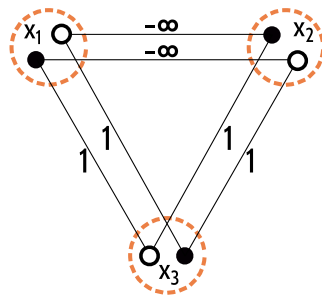


Graph $G = (V, E)$ with edge weight function w

- A boolean variable x_i per vertex $i \in V$
- A cost function per edge $e = (i, j) \in E : w_{ij} = w(i, j) \times \mathbb{1}[x_i \neq x_j]$
- Hard edges: constraints with costs 0 or $-\infty$ (when $x_i \neq x_j$)

3-clique

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



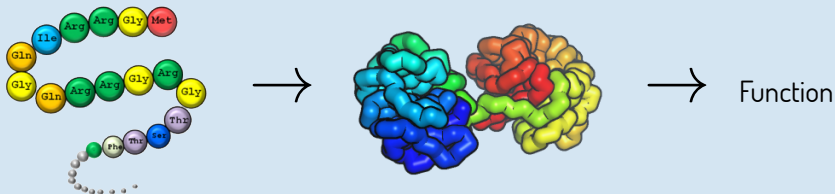
MAXCUT on a 3-clique with hard edge

```
{  
  "problem" :{"name": "MaxCut", "mustbe": ">0.0"},  
  "variables": {"x1": ["1","r"], "x2": ["1","r"], "x3": ["1","r"]},  
  "functions": {  
    "cut12": {"scope": ["x1","x2"], "costs": [0,-100,-100,0]},  
    "cut13": {"scope": ["x1","x3"], "costs": [0,1,1,0]},  
    "cut23": {"scope": ["x2","x3"], "costs": [0,1,1,0]}  
  }  
}
```

Most active molecules of life

Sequence of “amino-acids”, each chosen among a set of 20 natural ones

Folding



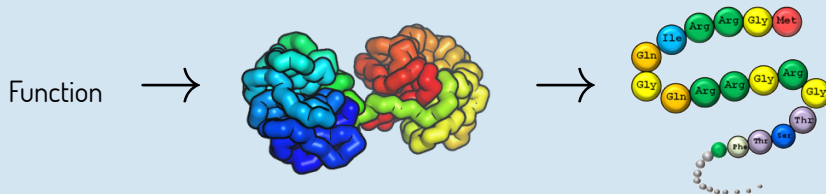
Transporter, binder, regulator, motor, catalyst...

Hemoglobine, TAL effector, ATPase, dehydrogenases...

Most active molecules of life

Sequence of “amino-acids”, each chosen among a set of 20 natural ones

Inverse folding



Transporter, binder, regulator, motor, catalyst...

Hemoglobine, TAL effector, ATPase, dehydrogenases...

New eco-friendly chemical/structural nano-agents

- Already produced new folds,² catalysts,⁵ nano-components⁸

New eco-friendly chemical/structural nano-agents

- Already produced new folds,² catalysts,⁵ nano-components⁸
- Useful for biomass transformation (biofuels, food and feed, cosmetics...),

New eco-friendly chemical/structural nano-agents

- Already produced new folds,² catalysts,⁵ nano-components⁸
- Useful for biomass transformation (biofuels, food and feed, cosmetics...),
- For new drugs in medicine

New eco-friendly chemical/structural nano-agents

- Already produced new folds,² catalysts,⁵ nano-components⁸
- Useful for biomass transformation (biofuels, food and feed, cosmetics...),
- For new drugs in medicine
- To provide new components for nanotechnologies

New eco-friendly chemical/structural nano-agents

- Already produced new folds,² catalysts,⁵ nano-components⁸
- Useful for biomass transformation (biofuels, food and feed, cosmetics...),
- For new drugs in medicine
- To provide new components for nanotechnologies

20^n sequences!

intractable for experimental techniques

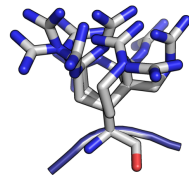
Ingredients

- Full atom model of a protein backbone

(assumed to be rigid)

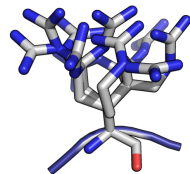
Ingredients

- Full atom model of a protein backbone (assumed to be rigid)
- Catalog of all 20 amino acids in different conformations (≈ 400 overall)



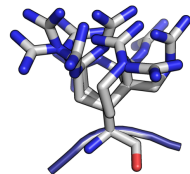
Ingredients

- Full atom model of a protein backbone (assumed to be rigid)
- Catalog of all 20 amino acids in different conformations (≈ 400 overall)
- Full atom energy function (bonds, electrostatics, solvent, statistics...)



Ingredients

- Full atom model of a protein backbone (assumed to be rigid)
- Catalog of all 20 amino acids in different conformations (≈ 400 overall)
- Full atom energy function (bonds, electrostatics, solvent, statistics...)
- Maximum stability \equiv Minimum energy NP-hard⁴

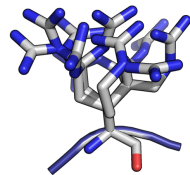


Ingredients

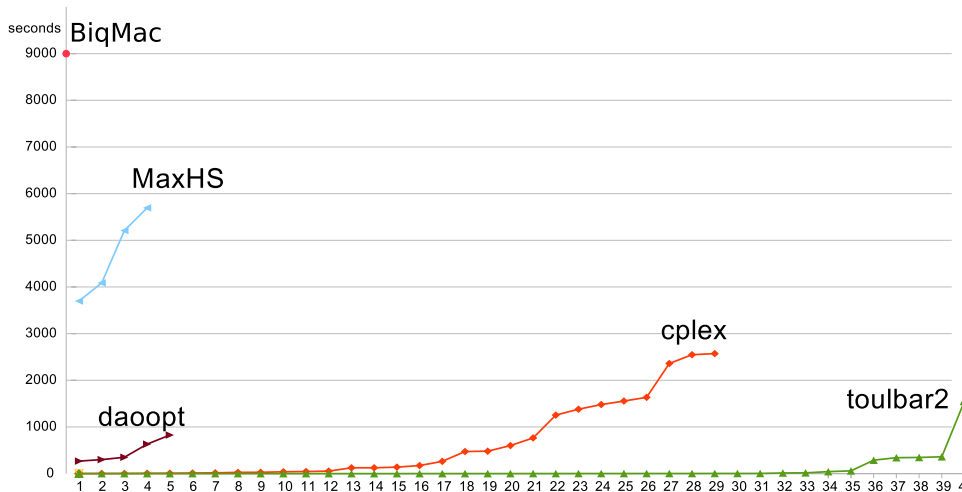
- Full atom model of a protein backbone (assumed to be rigid)
- Catalog of all 20 amino acids in different conformations (≈ 400 overall)
- Full atom energy function (bonds, electrostatics, solvent, statistics...)
- Maximum stability \equiv Minimum energy NP-hard⁴

As a Cost Function Network

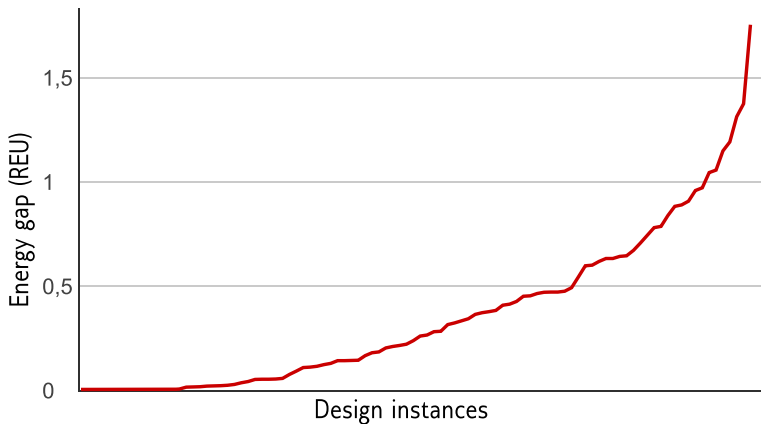
- One variable per position in the protein sequence
- Domain: catalog of few hundreds amino acids conformations
- Functions: decomposed energy (sum of pairwise terms)
- Search space has size $\approx 400^n$



Toulbar2 vs. CPLEX, MaxHS...(real instances)



of instances solved (X) within a per instance cpu-time limit (Y)



Optimality gap of the Simulated annealing solution as problems get harder
Asymptotic convergence can be arbitrarily slow

Imperfect

- Approximations: rigidity, solvent effect...
- Ignores: interactions inside the cell, polarisability...
- Needs more information, extracted from data

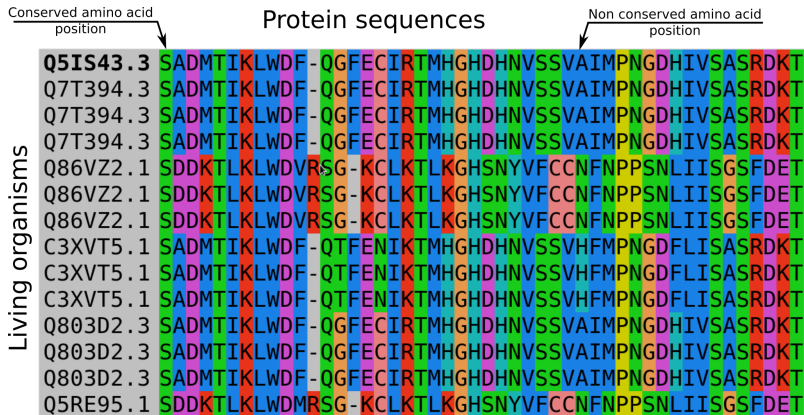
Imperfect

- Approximations: rigidity, solvent effect...
- Ignores: interactions inside the cell, polarisability...
- Needs more information, extracted from data

Evolutionary information

- Use similar proteins (homologs) from databases
- Multiple alignment: align similar regions of the sequences

A multiple alignment with conserved positions

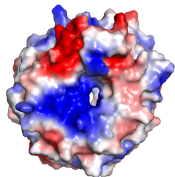
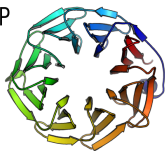


Simple integration of information


- Force amino acid choice (constraint) at conserved positions.

C8 pseudo-symmetric 20VP symmetrized into a nano-component

20VP

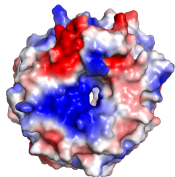
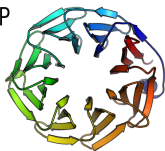


C8 pseudo-symmetric 20VP symmetrized into a nano-component

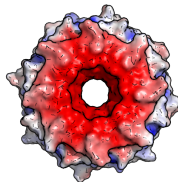
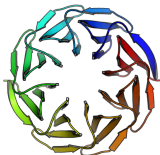
-  Tako: (R)evolution + Rosetta/talaris14

8 fold



20VP



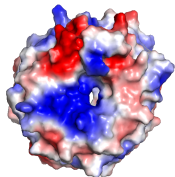
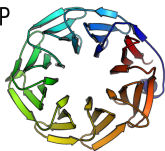
Tako



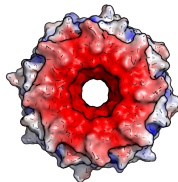
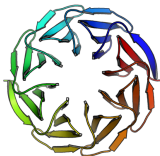
C8 pseudo-symmetric 20VP symmetrized into a nano-component

-  Tako: (R)evolution + Rosetta/talaris14 8 fold
-  Ika: toulbar2 + talaris14 4 fold

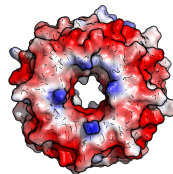
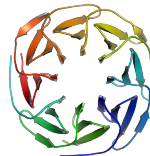
20VP

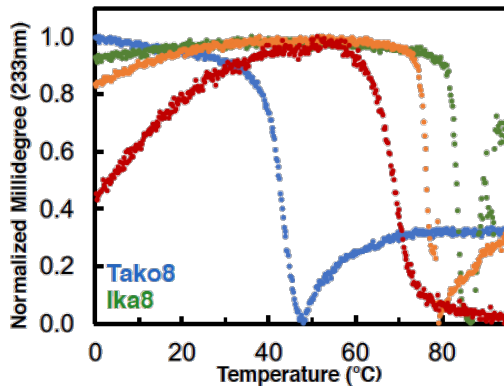


Tako



Ika





Compares Tako and Ika structural stability as temperature increases
(circular dichroism)

Boltzman distribution connects probability and cost

$$P(X) \propto e^{-W(X)}$$

Boltzman distribution connects probability and cost

$$P(X) \propto e^{-W(X)}$$

From CFN to probabilities and back

- After e^{-x} transform, a CFN defines a probability distribution (MRF)

Boltzman distribution connects probability and cost

$$P(X) \propto e^{-W(X)}$$

From CFN to probabilities and back

- After e^{-x} transform, a CFN defines a probability distribution (MRF)
- Which can be learned from data using maximum penalized likelihood [1, 3, 6]

Boltzman distribution connects probability and cost

$$P(X) \propto e^{-W(X)}$$

From CFN to probabilities and back

- After e^{-x} transform, a CFN defines a probability distribution (MRF)
- Which can be learned from data using maximum penalized likelihood [1, 3, 6]
- And transformed back into a CFN with a $-\log(x)$ transform

- We start from a complete pairwise CFN with unknown cost functions

- We start from a complete pairwise CFN with unknown cost functions
- We have a total of $d^2 \cdot \frac{n(n-1)}{2}$ parameters to learn

$w_{ij}(\cdot, \cdot)$

- We start from a complete pairwise CFN with unknown cost functions
- We have a total of $d^2 \cdot \frac{n(n-1)}{2}$ parameters to learn
- Let $\ell(D|w_{ij})$ be the log-probability of data D given the w_{ij}

$w_{ij}(\cdot, \cdot)$

- We start from a complete pairwise CFN with unknown cost functions
- We have a total of $d^2 \cdot \frac{n(n-1)}{2}$ parameters to learn
- Let $\ell(D|w_{ij})$ be the log-probability of data D given the w_{ij}

$w_{ij}(\cdot, \cdot)$

Maximize $\ell(D|w_{ij}) - \lambda \cdot ||w_{ij}||$

concave

- We start from a complete pairwise CFN with unknown cost functions
- We have a total of $d^2 \cdot \frac{n(n-1)}{2}$ parameters to learn
- Let $\ell(D|w_{ij})$ be the log-probability of data D given the w_{ij}

$w_{ij}(\cdot, \cdot)$

Maximize $\ell(D|w_{ij}) - \lambda \cdot ||w_{ij}||$

concave

Efficient L2 norm based implementation available [6]

- Uses conjugate gradient optimization
- fast C or very fast CUDA implementation
- n variables, d values, s samples: $O(d^2n^2 + dns)$ space.

- We start from a complete pairwise CFN with unknown cost functions
- We have a total of $d^2 \cdot \frac{n(n-1)}{2}$ parameters to learn
- Let $\ell(D|w_{ij})$ be the log-probability of data D given the w_{ij}

$w_{ij}(\cdot, \cdot)$

Maximize $\ell(D|w_{ij}) - \lambda \cdot ||w_{ij}||$

concave

Efficient L2 norm based implementation available [6]

- Uses conjugate gradient optimization
- fast C or very fast CUDA implementation
- n variables, d values, s samples: $O(d^2n^2 + dns)$ space.

600 variables, domain size 21

80,000,000 parameters, estimated in minutes

Let's recap...

- Model the problem as a CFN (generalizes CP)

Let's recap...

- Model the problem as a CFN (generalizes CP)
- Learn other CFNs from available data sets using penalized likelihood optimization

Let's recap...

- Model the problem as a CFN (generalizes CP)
- Learn other CFNs from available data sets using penalized likelihood optimization
- Combine the models by scaling/adding/connecting them together

Let's recap...

- Model the problem as a CFN (generalizes CP)
- Learn other CFNs from available data sets using penalized likelihood optimization
- Combine the models by scaling/adding/connecting them together
- Solve them with toulbar2

: -)

MIT licence, <https://github.com/toulbar2/toulbar2>

AI/toulbar2

S. de Givry (INRA)
G. Katsirelos (INRA)
M. Zytnicki (PhD, INRA)
D. Allouche (INRA)
H. Nguyen (PhD, INRA)
M. Cooper (IRIT, Toulouse)
J. Larrosa (UPC, Spain)
F. Heras (UPC, Spain)
M. Sanchez (Spain)
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)

Protein Design

A. Voet (KU Leuven)
D. Simoncini (INSA, Toulouse)
S. Barbe (INSA, Toulouse)
S. Traoré (PhD, CEA)
C. Viricel (PhD)
PyRosetta (U. John Hopkins)
OSPREY (Duke U.)

- [1] Sivaraman Balakrishnan et al. "Learning generative models for protein fold families". In: *Proteins: Structure, Function, and Bioinformatics* 79.4 (2011), pp. 1061–1078.
- [2] Brian Kuhlman et al. "Design of a novel globular protein fold with atomic-level accuracy". In: *science* 302.5649 (2003), pp. 1364–1368.
- [3] Youngsuk Park et al. "Learning the Network Structure of Heterogeneous Data via Pairwise Exponential Markov Random Fields". In: *Artificial Intelligence and Statistics*. 2017, pp. 1302–1310.
- [4] Niles A Pierce and Erik Winfree. "Protein design is NP-hard.". In: *Protein Eng.* 15.10 (Oct. 2002), pp. 779–82. ISSN: 0269–2139. URL: <http://www.ncbi.nlm.nih.gov/pubmed/12468711>.
- [5] Daniela Röthlisberger et al. "Kemp elimination catalysts by computational enzyme design". In: *Nature* 453.7192 (2008), p. 190.
- [6] Stefan Seemayer, Markus Gruber, and Johannes Söding. "CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations". In: *Bioinformatics* 30.21 (2014), pp. 3128–3130.
- [7] David Simoncini et al. "Guaranteed Discrete Energy Optimization on Large Protein Design Problems". In: *Journal of Chemical Theory and Computation* 11.12 (2015), pp. 5980–5989. DOI: 10.1021/acs.jctc.5b00594.
- [8] Arnout RD Voet et al. "Computational design of a self-assembling symmetrical β -propeller protein". In: *Proceedings of the National Academy of Sciences* 111.42 (2014), pp. 15102–15107.