Constraint Programming approaches to the Protein Folding Problem.

Agostino Dovier
DIMI, University of Udine (IT)
www.dimi.uniud.it/dovier
www.dimi.uniud.it/dovier/PF

Outline of the talk

- Basic notions on Proteins
- Introduction to *Protein Folding/Structure Prediction Problem*
- The PFP as a *constrained optimization problem* ($CLP(\mathcal{FD})$)
  - Abstract modeling (HP) and solutions
  - Realistic modeling and solutions
- Simulation approach using *Concurrent Constraint Programming*.
- Other approaches
- Conclusions

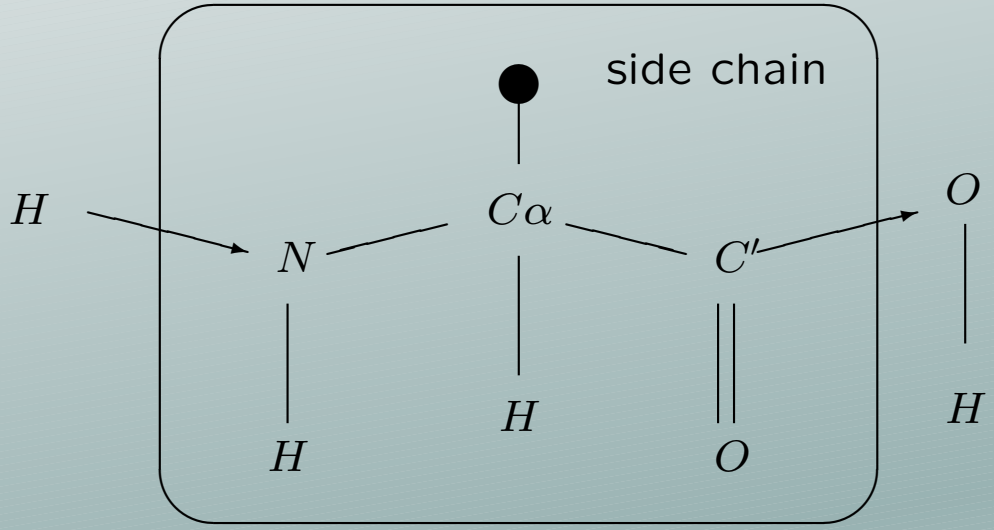
Proteins

- Proteins are abundant in all organisms and fundamental to life.
- The diversity of 3D protein structure underlies the very large range of their function:
  - Enzymes—biological catalysts
  - Storage (e.g. ferritin in liver)
  - Transport (e.g. haemoglobin)
  - Messengers (transmission of nervous impulses—hormones)
  - Antibodies
  - Regulation (during the process to synthesize proteins)
  - Structural proteins (mechanical support, e.g. hair, bone)

Primary Structure

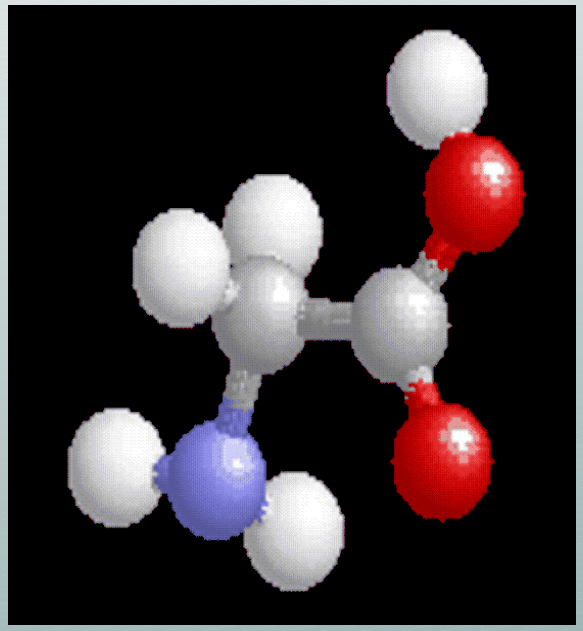
- A Protein is a polymer chain (a *list*) made of monomers (*aminoacids*).
- This list is called the *Primary Structure*.
- The typical length is 50–500.
- Aminoacids are of twenty types, called **Ala**nine (A), **Cys**teine (C), **Asp**artic Acid (D), **Glu**tamic Acid (E), **Phe**nylalanine (F), **Gly**cine (G), **His**tidine (H), **I**so**le**ucine (I), **Lys**ine (K), **Leu**cine (L), **Met**hionine (M), **As**paragi**n**e (N), **Pro**line (P), **Gl**utami**n**e (Q), **Arg**inine (R), **Ser**ine (S), **Thr**eonine (T), **Val**ine (V), **Tryp**tophan (W), **Tyr**osine (Y).
- *Summary:* The primary structure of a protein is a list of the form $[a_1, \ldots, a_n]$ with $a_i \in \{A, \ldots, Z\} \setminus \{B, J, O, U, X, Z\}$.

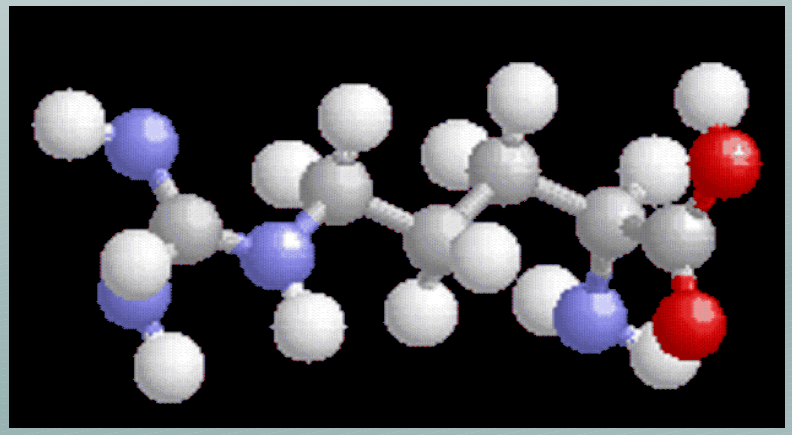
Aminoacid Structure



- The *backbone* is the same for all aminoacids.
- The *side chain* characterizes each aminoacid.
- Side chains contain from 1 (Glycine) to 18 (Tryptophan) atoms.

Example: Glycine and Arginine





$C_2H_5NO_2 \rightarrow$ 10 atoms

$C_6H_{14}N_4O_2 \rightarrow$ 26 atoms

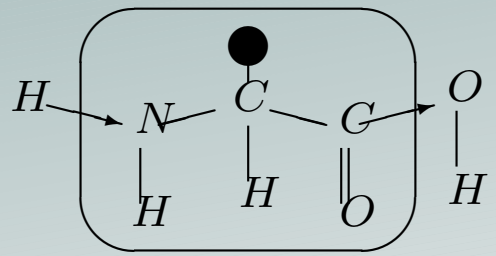
Remember the basic scheme (9 atoms) $\Rightarrow$

White $=$ H

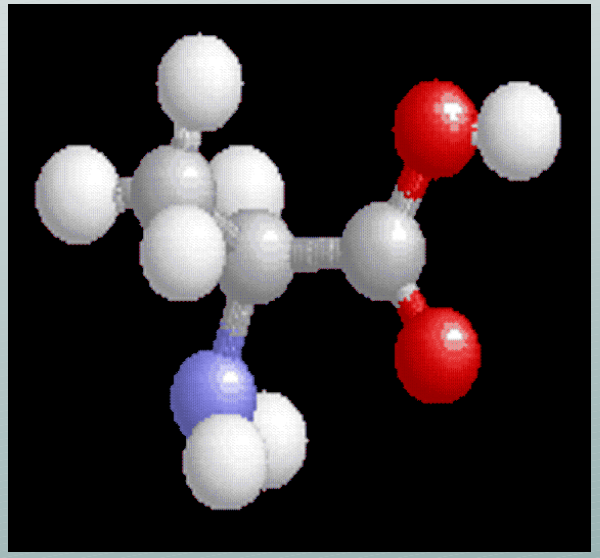
Blue $=$ N

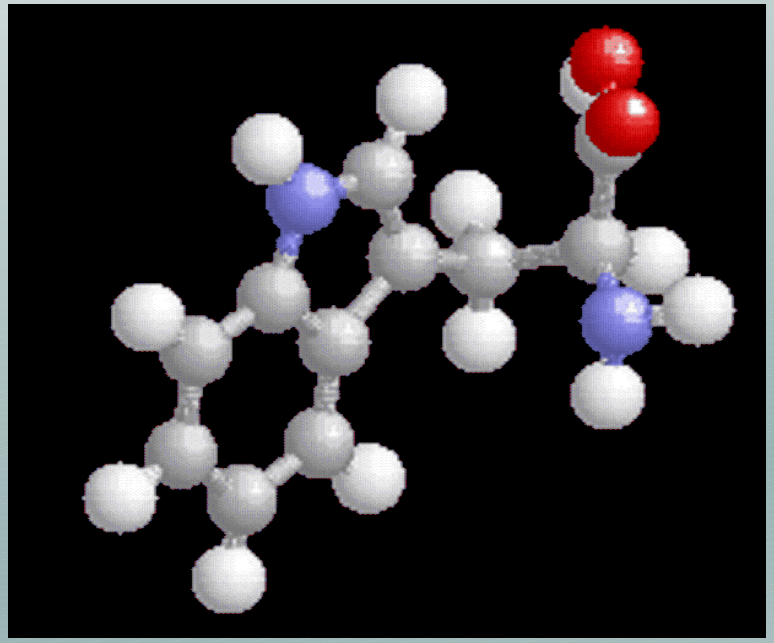
Red $=$ O

Grey $=$ C



Example: Alanine and Tryptophan

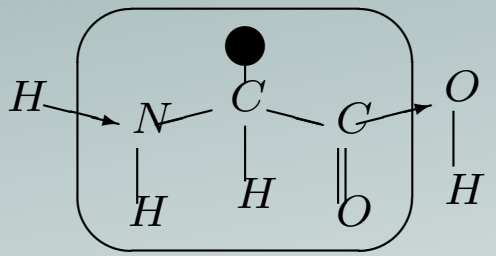




$C_3H_7NO_2 \rightarrow 13$ atoms

$C_{11}H_{12}N_2O_2 \rightarrow 27$ atoms

White $=$ H
Blue $=$ N
Red $=$ O
Grey $=$ C



Aminoacid's size

Name	Chemical	Side Chain
A	$C_3H_7NO_2$	4
C	$C_3H_7NO_2S$	4
D	$C_4H_7NO_4$	16
E	$C_5H_9NO_4$	10
F	$C_9H_{11}NO_2$	14
G	$C_2H_5NO_2$	1
H	$C_6H_9N_3O_2$	11
I	$C_6H_{13}NO_2$	13
K	$C_6H_{14}N_2O_2$	15
L	$C_6H_{13}NO_2$	13

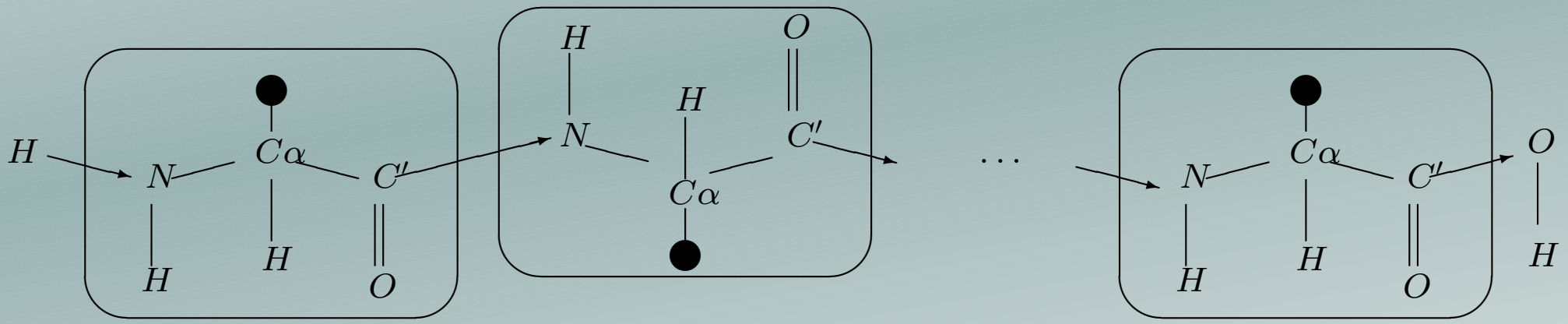
Name	Chemical	Side Chain
M	$C_5H_{11}NO_2S$	11
N	$C_4H_8N_2O_3$	8
P	$C_5H_9NO_2$	8(*)
Q	$C_5H_{10}N_2O_3$	11
R	$C_6H_{14}N_4O_2$	17
S	$C_3H_7NO_3$	5
T	$C_4H_9NO_3$	9
Y	$C_9H_{11}NO_3$	15
V	$C_5H_{11}NO_2$	10
W	$C_{11}H_{12}N_2O_2$	18

Images from:

`http://www.chemie.fu-berlin.de/chemistry/bio/amino-acids_en.html`

Primary Structure, detailed

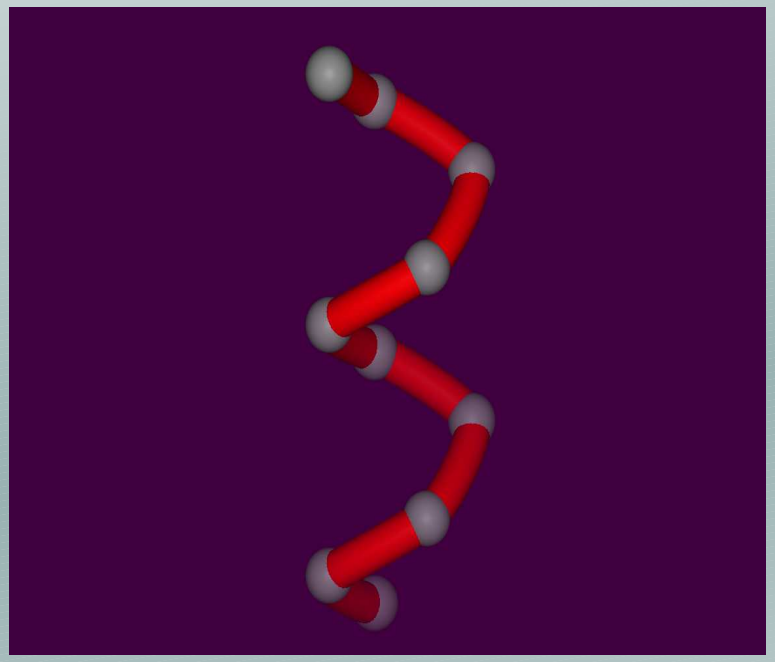
- The primary structure is a linked list of aminoacids.
- The terminals H (left) and OH (right) are lost in the linking phase.

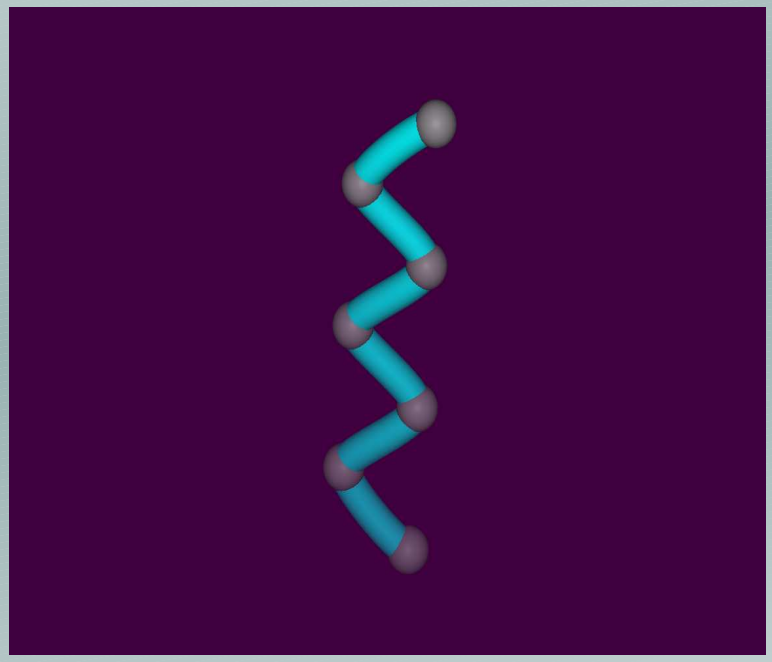


The Secondary Structure

- Locally, a protein *can* assume two particular forms:

  α-helix *β-sheet*





- This kind of information is called the *Secondary Structure* of a Protein.

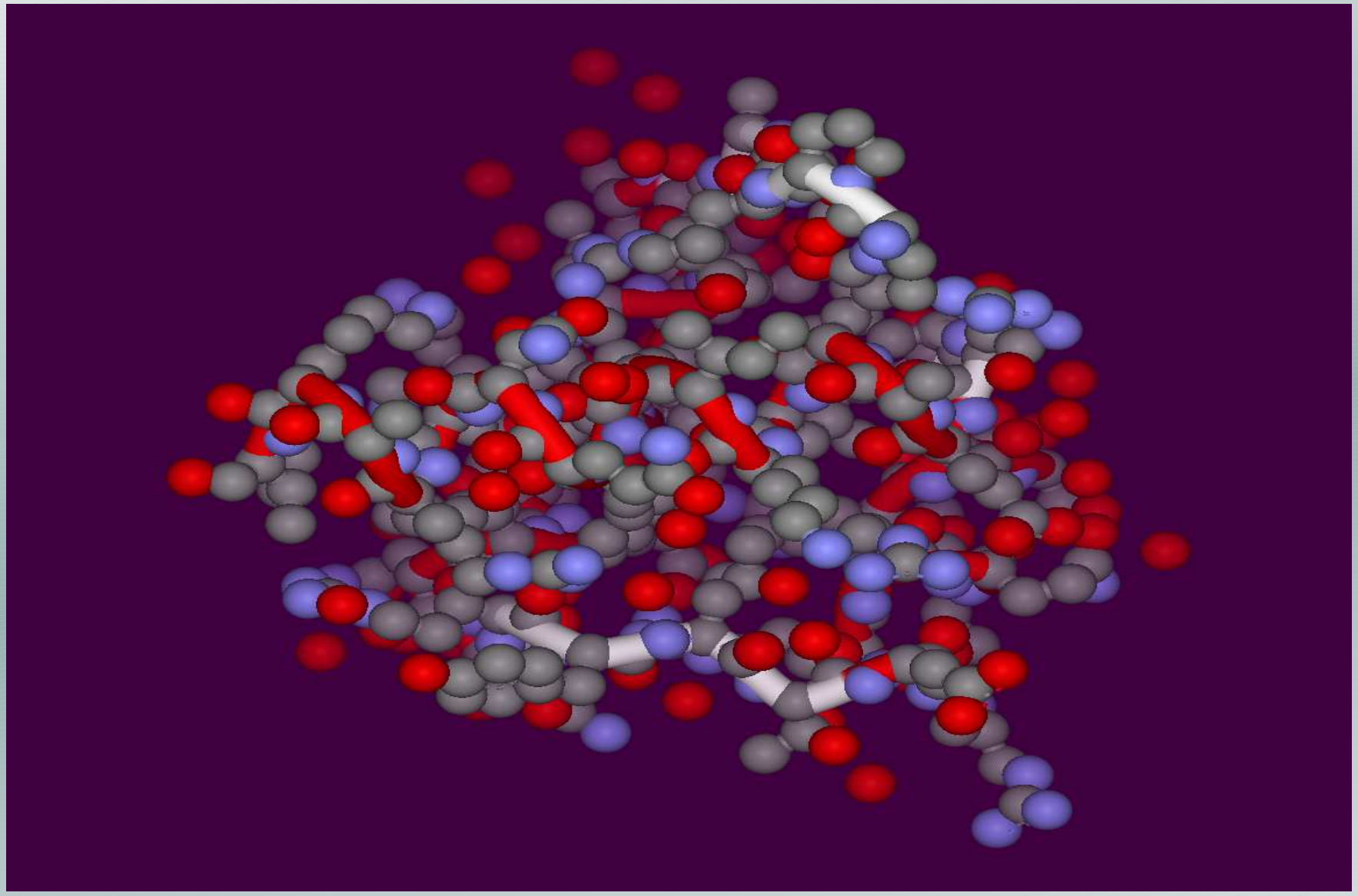
The Tertiary Structure

- The complete 3D conformation of a protein is called the *Tertiary Structure*.
- Proteins *fold* in a determined environment (e.g. water) to form a very specific geometric pattern (*native state*).
- The native conformation is relatively stable and unique and (*Anfinsen*'s hypothesis) is the state with minimum free energy.
- The tertiary structure determines the *function* of a Protein.
- $\sim$ 29500 structures (most of them redundant) are stored in the PDB.
- The number of possible proteins of length ≤ 500 is

$$20^1 + 20^2 + \cdots + 20^{500} = O(20^{501}) \sim 10^{651}$$

- It is supposed that the secondary structures form before the tertiary.

Example: Tertiary Structure of 1ENH



The Protein Folding Problem

- The *Protein Structure Prediction (PSP) problem* consists in predicting the Tertiary Structure of a protein, given its Primary Structure.
- The *Protein Folding (PF) Problem* consists in predicting the whole folding process to reach the Tertiary Structure.
- Sometimes the two problems are not distinguished.
- A reliable solution is fundamental for medicine, agriculture, Industry.
- Let us focus on the PSP problem, first.

The PSP Problem

- Anfinsen: the native state minimizes the whole protein *energy*. Two problems emerge.

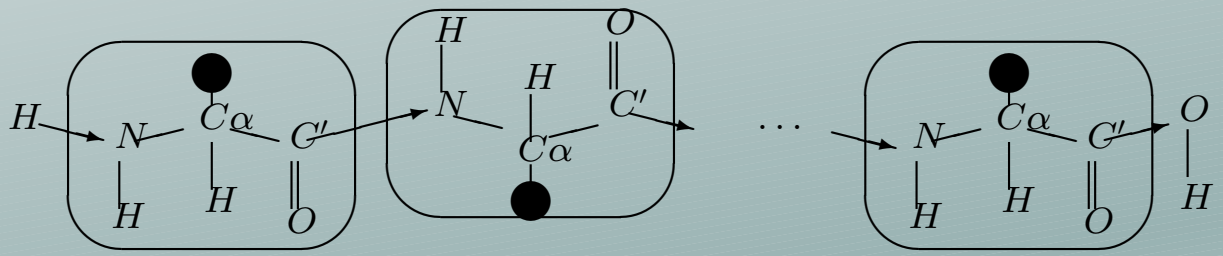
1 Energy model:
   - What is the energy function $\mathbb{E}$?
   - It depends on what?

2 Spatial Model: Assume $\mathbb{E}$ be known, depending on the aminoacids $a_1, \ldots, a_n$ and on their positions, what is the search's space where looking for the conformation minimizing $\mathbb{E}$?
   - Lattice (discrete) models.
   - Off-lattice (continuous) models.

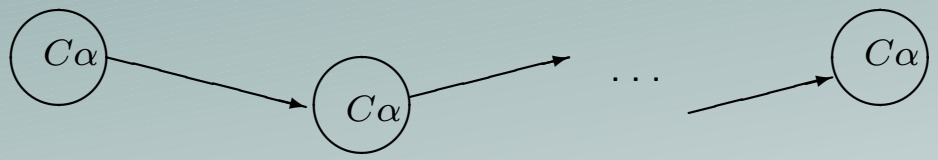
- After a choice for (1) and (2), we can face the minimization problem

A first assumption

- It emerges from experiments on the known proteins, that the distance between two consecutive $C\alpha$ atoms is fixed (3.8Å).

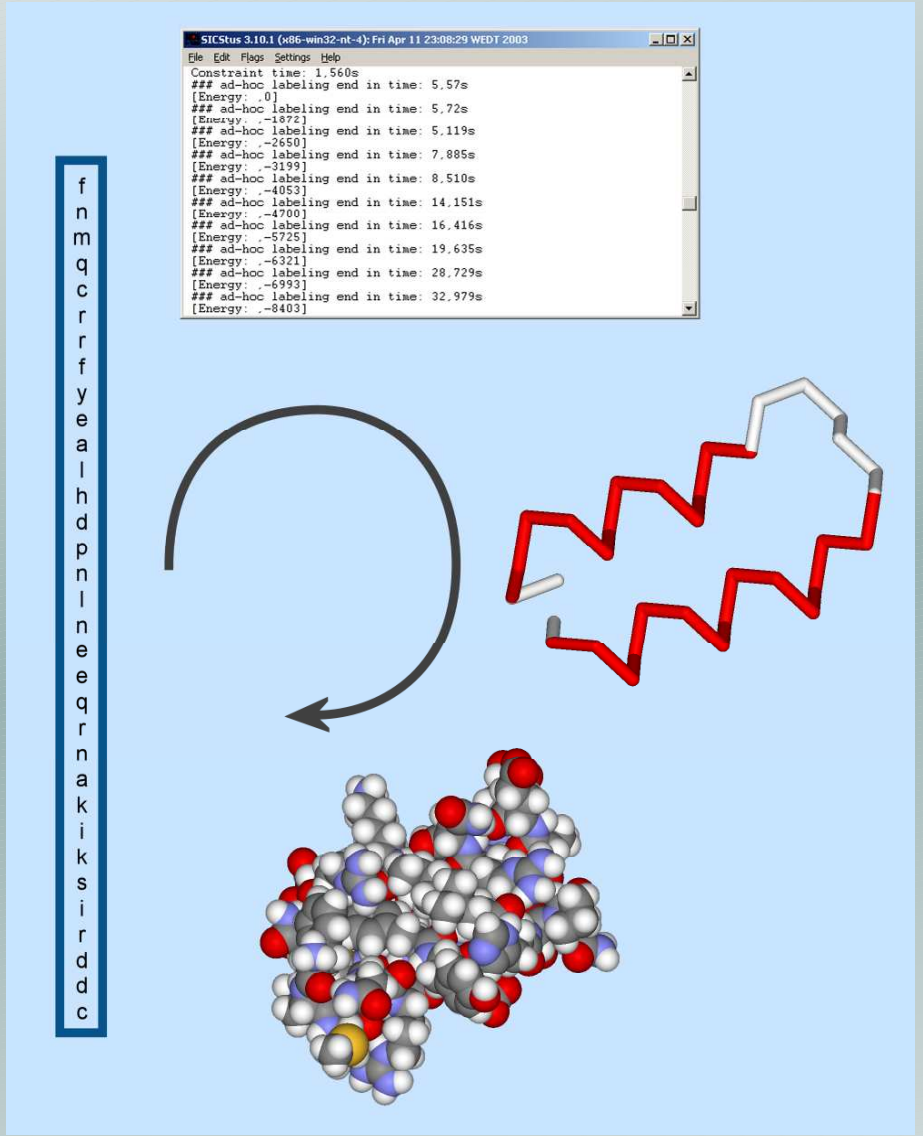


- We consider each aminoacid as a whole: *a box of fixed size centered in its $C\alpha$-atom*.



- The distance between two consecutive $C\alpha$ is chosen as unitary.

Protein Structure Prediction



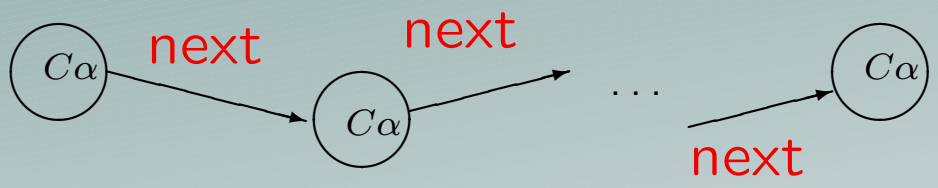
The general notion of folding

- Let $\mathcal{L}$ be the set of admissible points for each aminoacid.
- Given the sequence $a_1 \dots a_n$, a *folding* is a function

$$\omega : \{1, \dots, n\} \longrightarrow \mathcal{L}$$

  such that:
  - next$(\omega(i), \omega(i+1))$ for $i = 1, \dots, n-1$, and
  - $\omega(i) \neq \omega(j)$ for $i \neq j$.

- *next* forces the unitary distance between consecutive aminoacids.



Objective function

- *Assumption*: the energy is the sum of the energy contributions of each pair of non-consecutive aminoacids.
- It depends on their *distance* and on their *type*. The contribution is of the form en_contrib(ω, i, j).
- The function to be minimized is therefore:

$$E(\omega) = \sum_{\substack{1 \leq i \leq n \\ i+2 \leq j \leq n}} \text{en\_contrib}(\omega, i, j)$$

- It is a constrained minimization problem (recall that: next$(\omega(i), \omega(i+1))$ and $\omega(i) \neq \omega(j)$).
- It is parametric on $\mathcal{L}$, next, and en_contrib.
- next and en_contrib are typically non linear.

A first proposal for the Energy: DILL

- The aminoacids: Cys (C), Ile (I), Leu (L), Phe (F), Met (M), Val (V), Trp (W), His (H), Tyr (Y), Ala (A) are *hydrophobic* (H).
- The aminoacids: Lys (K), Glu (E), Arg (R), Ser (S), Gln (Q), Asp (D), Asn (N), Thr (T), Pro (P), Gly (G) are *polar* (P).
- The protein is in water: hydrophobic elements tend to occupy the center of the protein.
- Consequently, H aminoacids tend to stay close each other.
- polar elements tend to stay in the frontier.

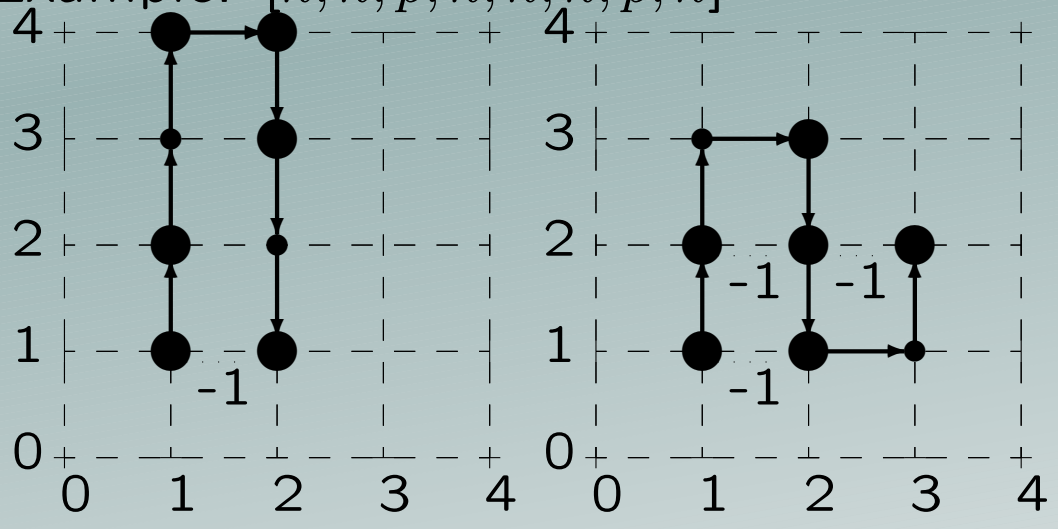
A first proposal for the Energy: DILL

- This fact suggest an energy definition:
  - If two aminoacids of type H are *in contact* (i.e. no more distant than a certain value) in a folding they contribute negatively to the energy.
  - Otherwise their contribution is zero.

- The notion of being *in contact* is naturally formalized in *lattice models*: typically one *lattice unit*.

The simplest PFP formalization

- The spatial model is a subset of $\mathbb{N}^2$.
- A *contact* is when $|X_1 - X_2| + |Y_1 - Y2| = 1$.
- The primary list is a sequence of h and p.
- Each contact between pairs of h contributes as -1.
- We would like to find the folding(s) minimizing this energy
- Example: $[h, h, p, h, h, h, p, h]$



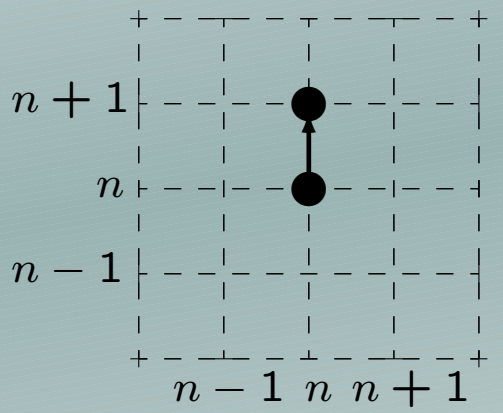
Unfortunately, the decision version:
Is there a folding with Energy $< k$?
is *NP-complete*

HP on $\mathbb{N}^2$

- If the primary structure is $[a_1, \ldots, a_n]$ with $a_i \in h, p$, then

$$\omega(i) \in \mathcal{L} = \{(i, j) : i \in [1..2n-1], j \in [1..2n-1]\}$$

- We can assume that:
  $\omega(1) = (n, n)$.
- and, to avoid symmetries, that:
  $\omega(2) = (n, n+1)$.



- We need to implement *next*, *en_contrib*, ...

- Let us see a simple (and working) $CLP(\mathcal{FD})$ code.

Encoding the PF, HP, on $\mathbb{N}^2$

```
pf(Primary, Tertiary) :-        %%% Primary = [a1,...,aN], ai in {h,p}
        constrain(Primary,Tertiary,Energy),
        labeling([ff,minimize(Energy)],Tertiary).

constrain(Primary,Tertiary,Energy) :-
        length(Primary,N),
        M is 2*N, M1 is M - 1,
        length(Tertiary,M),  %%% Tertiary = [X1,Y1,...,XN,YN]
        domain(Tertiary,1,M1),
        starting_point(Tertiary,N),
        avoid_loops(Tertiary),
        next_constraints(Tertiary),
        energy_constraint(Primary,Tertiary,Energy).
```

Encoding the PF, HP, on $\mathbb{N}^2$

```
starting_point([N,N,N,N1|_],N) :-  %%% X1=Y1=X2=N, Y2=N+1
      N1 is N + 1.

avoid_loops(Tertiary):-
       positions_to_integers(Tertiary, ListaInteri),
       all_different(ListaInteri).

positions_to_integers([X,Y|R], [I|S]):-
       I  #= X*100+Y,    %%% 100 is a "large" number
       positions_to_integers(R,S).
positions_to_integers([],[]).
```

This way, we do not introduce a disjunction

$$X_i \neq X_j \vee Y_i \neq Y_j$$

for each constraint

$$(X_i, Y_i) \neq (X_j, Y_j)$$

Encoding the PF, HP, on $\mathbb{N}^2$

```
next_constraints([_,_]).
next_constraints([X1,Y1,X2,Y2|C]) :-
        next(X1,Y1,X2,Y2),
        next_constraints([X2,Y2|C]).

next(X1,Y1,X2,Y2):-
        domain([Dx,Dy],0,1),
        Dx #= abs(X1-X2),
        Dy #= abs(Y1-Y2),
        Dx + Dy #= 1.
```

Note: a non linear constraint.

Encoding the PF, HP, on $\mathbb{N}^2$

```
energy_constraint(Primary,Tertiary,Energy):- ...
```

is defined recursively so as to fix

$$\texttt{Energy \#=} \; C_{1,3} + C_{1,4} + \cdots + C_{1,N} + C_{2,4} + \cdots + C_{2,N} + \vdots + C_{N-2,N}$$

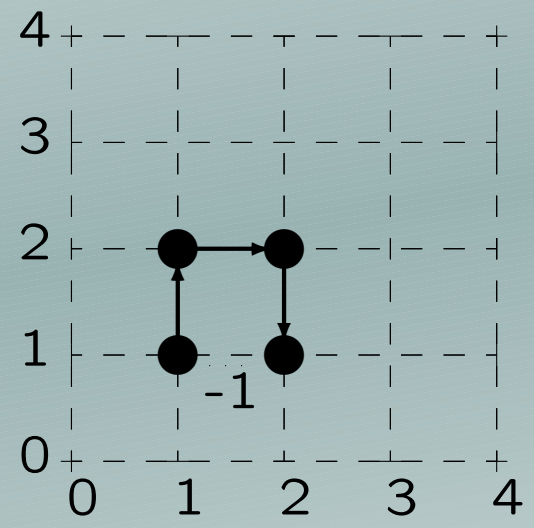
Where each $C_{A,B}$ is defined as follows:

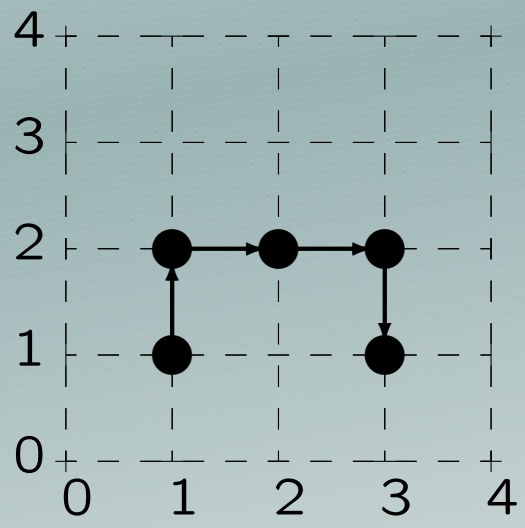
```
energy(h,XA,YA,h,XB,YB,C_AB) :-
       C_AB in {0,-1},
       DX #= abs(XA - XB),
       DY #= abs(YA - YB),
       1 #= DX + DY  #<=> C_AB #= -1.
energy(h,_,_,p,_,_,0). energy(p,_,_,h,_,_,0). energy(p,_,_,p,_,_,0).
```

Note: a non linear constraint.

Constraint Optimization

- This basic code is a good starting point for Optimization.
- A first idea concerns the objective function `Energy`.
- Only aminoacids at an *odd* relative distance can contribute to the Energy.





- Proof: think to the offsets at each step.

Constraint Optimization

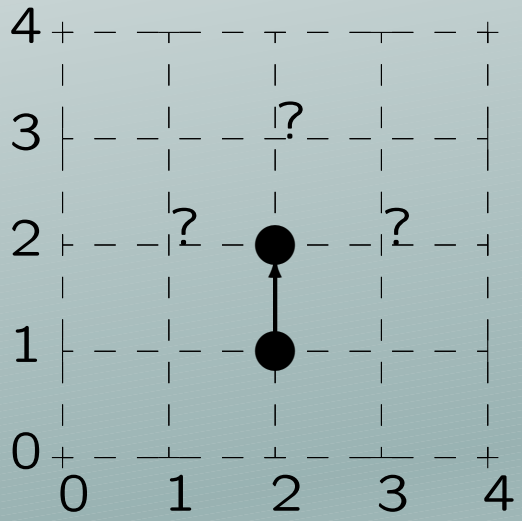
- Thus,

$$\texttt{Energy \#=} \underbrace{C_{1,3}}_{=0} + C_{1,4} + \underbrace{C_{2,4}}_{=0} + \cdots + C_{1,N} +$$
$$\underbrace{C_{2,4}}_{=0} + C_{2,5} + \cdots + C_{2,N} +$$
$$\vdots$$
$$+ \underbrace{C_{N-2,N}}_{=0}$$

- Speed up 3×.

Constraint Optimization

- The next element can (at the beginning) take 3 positions.
- Space looks as $\sim 3^n$
- More precisely, it is $\sim 1.28\frac{2.64^n}{n^{0.34}}$.
- We can reduce it by reducing the domains and/or adding linear distance constraints on pairs of aminoacids.



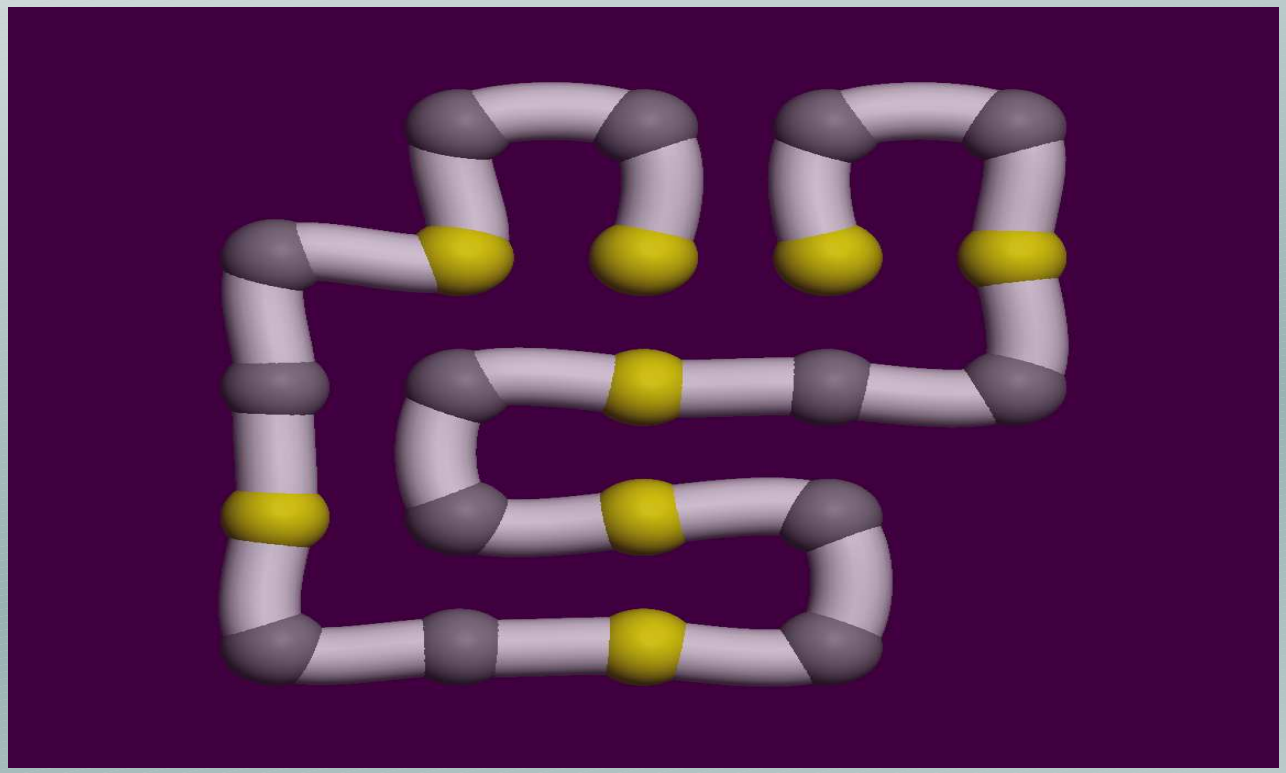
- Typically, in the solutions, the offsets are of the order of $2\sqrt{N}$ (for real proteins there are some more precise formulae).

- Speed up 20×.

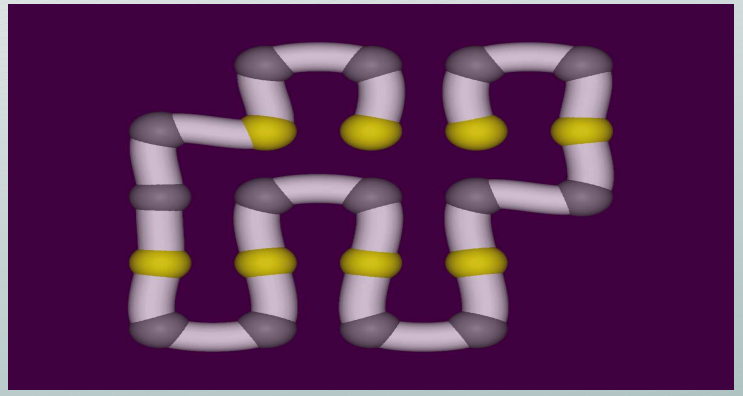
- Further speed up? Easy exercise :-) — hint: try approximated search e.g., `lds` strategy implemented in ECLiPSe.

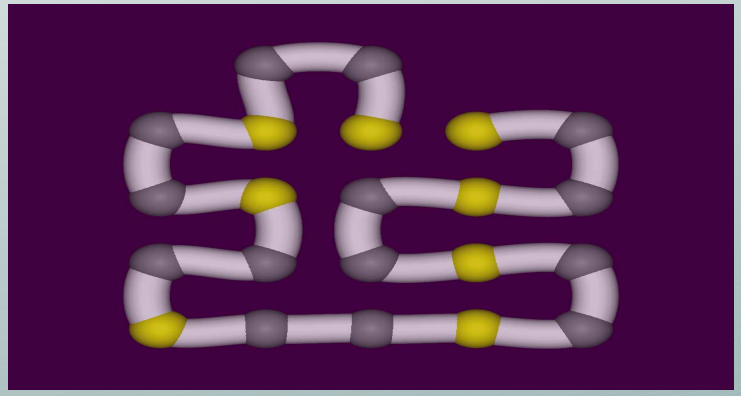
Example

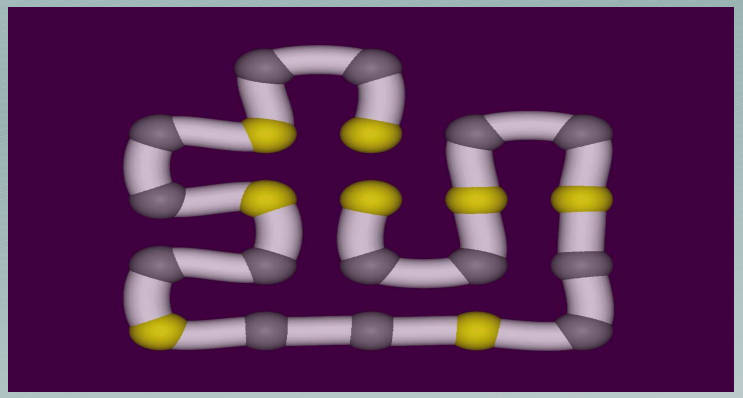
`:-pf([h,p,p,h,p,p,...,h,p,p,h],L)`, $n = 22$. 48 s., Energy = -6.

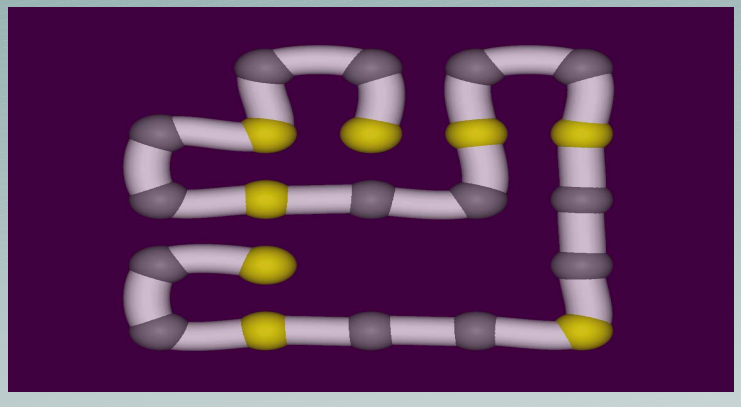


Example









Why not using Answer Set Programming?

Input	N	Min	SICStus	ECLiPSe	ASP Time	ASP rules
h^*	10	-4	0.2s	1.3s	2.0s	12132
h^*	15	-8	2.7s	21.2s	1m15s	47533
h^*	20	-12	4m41s	40m56s	3h34m	130018
h^*	25	-16	9h26m	91h44m	∞	205176
$(hpp)^*h$	10	-4	0.1s	0.3s	1.9s	12120
$(hpp)^*h$	16	-6	0.4s	4.2s	1m02s	54267
$(hpp)^*h$	22	-6	1m27s	9m19s	2h58m	157951
$(hpp)^*h$	28	-9	2h38m	18h32m	∞	258326
				$\sim$ 10x	$\sim$ 100x	

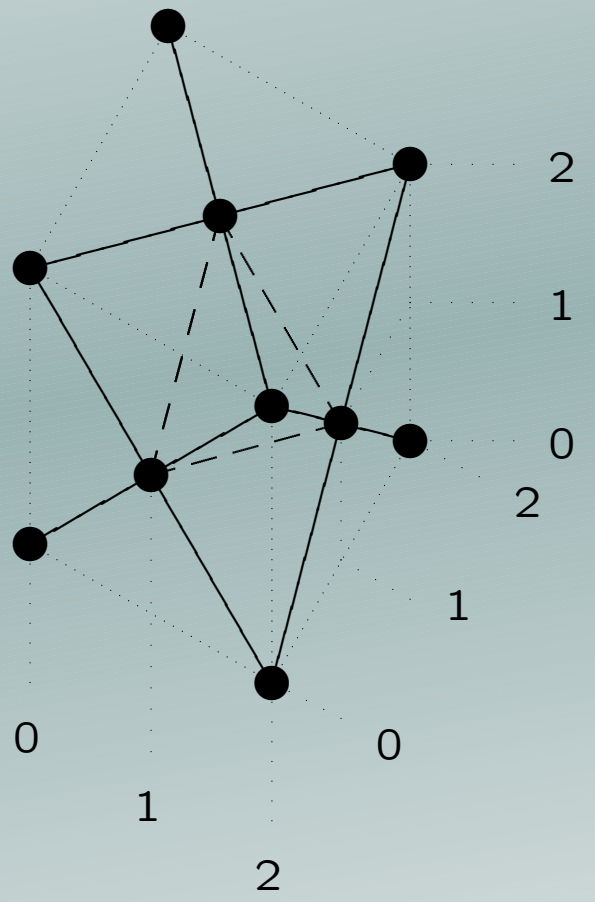
Problems of the HP model on $\mathbb{N}^2$ (or $\mathbb{N}^3$)

- Conformations are not realistic
- Too many equivalent minima (in nature proteins fold deterministically)
- Approximated search leads to useless conformations
- We need to consider other
  - space models
  - energy models

A more realistic space model

- The *Face Centered Cube (fcc) lattice* models the discrete space in which the protein can fold.

- It is proved to allow realistic conformations.
- The cube has size 2.
- Points such that $x+y+z$ is even are chosen.
- Points at distance $\sqrt{2}$ are connected
- Each point has 12 neighbors and $60°, 90°, 120°$ and $180°$ bend angles are allowed (in nature $60°$ and $180°$ never occur).



HP on FCC: Main Results

- A $Constraint(\mathcal{FD})$ program in Mozart by Backofen-Will folds HP-proteins up to length 150!
- Clever propagation, an idea of stratification and some geometrical results on the lattice.
- Drawbacks: It is only an abstraction. The solutions obtained are far from reality. For instance, helices and sheets are never obtained.
- Problems:
  - Energy function too simple.
  - Notion of contact.

A more realistic Energy function

- Same assumption: only pairs of aminoacids in *contact* contribute to the energy value.
- There is a *potential matrix* storing the contribution for each pair of aminoacids in contact.
- Values are either positive or negative.
- The global energy must be minimized.

PF, 20 aminoacids, on $\mathbb{N}^2$

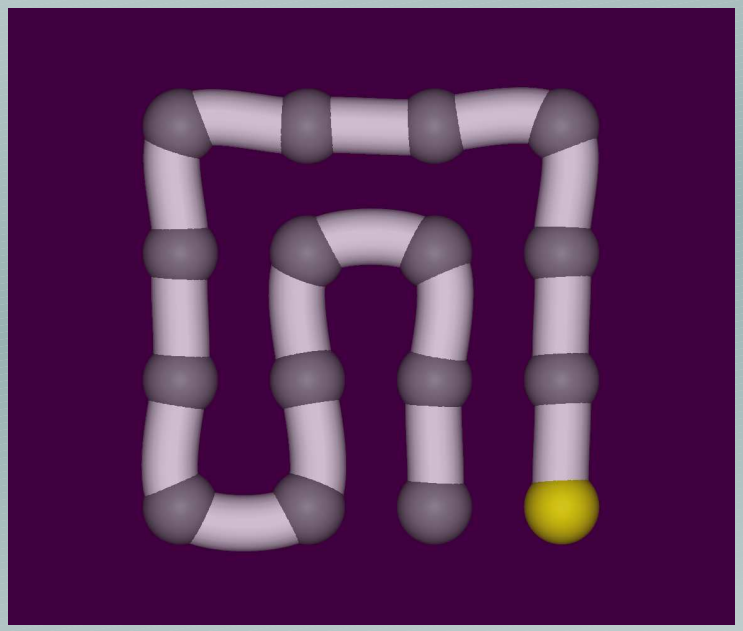
- Basically, the same Prolog code, with a call to `table(A,B,Cost)`.

```
energy(A,XA,YA,B,XB,YB,C) :-
        table(A,B,Cost),
        (Cost #\= 0,!,
        C in {0,Cost},
        DX #= abs(XA - XB),
        DY #= abs(YA - YB),
        1 #= DX + DY  #<=> C #= Cost;
        C #= 0).
```

- Potentials by Kolinsky and Skolnick.
- Or by Miyazawa and Jernigan,
- refined by Berrera, Fogolari, and Molinari.

Example

`:- pf([s,e,d,g, d,l,p,i, v,a,s,f, m,r,r,d],L).`, $n = 16$.
0.9 s., Energy = -7.4. (search space: 6.416.596)



PF, 20 aminoacids, on FCC

- The solution's space is huge $\sim 1.26n^{0.162}(10,03)^n$.
- Avoiding uncommon angles it is $\sim 5^n$
- The potential table is 20×20.
- Contact is set when $|X_1 - X_2| + |Y_1 - Y_2| + |Z_1 - Z_2| = 2$
- New constraints from secondary structure prediction (helices, strands, bonds) are needed.
- A careful treatment of the energy function using a matrix that statically set to 0 most of the elements $C_{i,j}$ defining the energy.
- E.g. if two aminoacids s_i, s_j belong to the same α-helix, the contribution of $C_{i,j}$ to the energy will be the same in any conformation: we set it to 0.

Using the Secondary Structure

- From an information of the kind helix(i, j) or strand(i, j) we can add strong distance constraints between variables

$$(X_i, Y_i, Z_i), (X_{i+1}, Y_{i+1}, Z_{i+1}), \ldots, (X_j, Y_j, Z_j)$$

- Secondary structure can be predicted with high accuracy: we can use these constraints.

- Moreover, *ssbonds* are induced by aminoacids: **Cys**teine ($C_3H_7NO_2S$) and **Met**hionine ($C_5H_{11}NO_2S$).

- A ssbond constrains the two aminoacids involved to be close in the Tertiary Structure: $|X_i - X_j| + |Y_i - Y_j| + |Z_i - Z_j| \leq 6$.

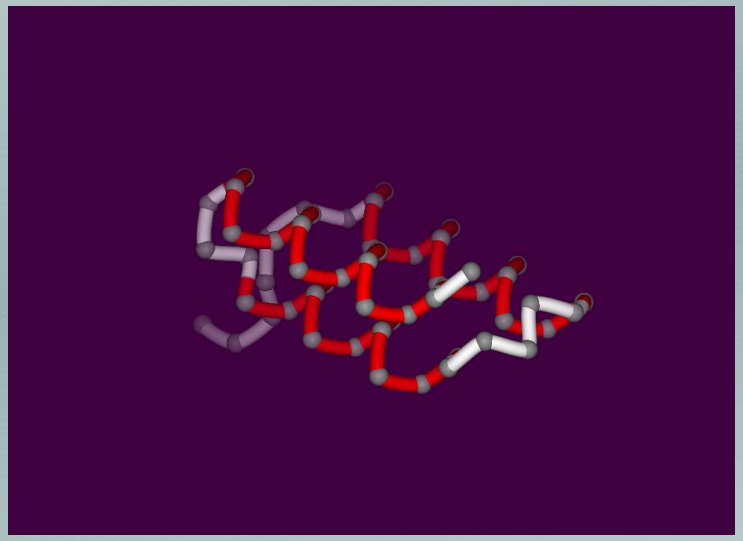
Labeling heuristics

- In the *labeling* stage,variables are assigned to values of their domains to find feasible solutions.
- A search tree is built, pruned by constraints.
- We use heuristics to select the next variable to be assigned.
- We use heuristics that cut the search if after some assignment the predicted value is not comparable with the minimum already obtained.
- We set a time limit to stop execution obtaining anyway a solution.

Example

- Proteins of length 60 can be predicted in some hours, with acceptable RMSD.

```
protein('1ENH', Primary, Secondary):-
   Primary = [r,p,r,t,a,f,s,s,e,q,
              l,a,r,l,k,r,e,f,n,e,
              n,r,y,l,t,e,r,r,r,q,
              q,l,s,s,e,l,g,l,n,e,
              a,q,i,k,i,w,f,q,n,k,
              r,a,k,i],
   Secondary = [helix(8,20),helix(26,36),
                helix(40,52), strand(22,23)].
```



Minimization updated results

- Wrt the version published in BMC Bioinformatics in Nov. 2004 we have now a sequential speed up of 10x ("cleaning" of secondary structure constraints and a new, promising, heuristic)

- New improvement distributing solution's search among several processors. This kind of parallelism is natural starting from Prolog code.
  We have obtained a speed up of 8–10x using a 9 processors of a parallel machine.

- Thus, 10x per month. The solution's space grows $\sim 5^n$. We don't want to solve $P = NP$ but to solve problems with $n = 150$. With some patience, we can work it out...

END of PART 1

Constraint-based Simulation Approach

- Molecular dynamics analyzes atom-atom interactions and computes forcefields.
- Then uses the forcefield for a global simulation move.
- The number of atoms is huge (7–24 per aminoacid) and computations involve solutions of differential equations.
- There are working tools, but detailed simulations of real-size proteins are not yet applicable.
- Moreover, it is easy to fall into local minima, and
- It seems that the folding follows more macroscopical laws.

Concurrent Constraint Simulation Approach

- *Idea:* perform simulations at higher abstraction level (aminoacids) using concurrent constraint programming.
- Basically, each aminoacid is an independent agent that communicates with the others.
- Motion follows some rules, governed by energy.
- Here we need more complex energy models (I will not enter into details).
- Off-lattice spatial model.
- We deal with the *protein folding problem*

Simulation

- Each agent waits for the communication of a move of some other aminoacid.
- He retrieved the the positions of all (or some of—depending on the chosen communication strategy) other aminoacids.
- It moves or it choose to keep still (see next slide).
- If it moves, it communicates to all (or some of) other agents its new position.

Moving Strategy

Each aminoacid performs a move in the following way:

- It randomly chooses a new position, close to the current one within a given range (a sphere of 0.2 Å).
- Using the most recent information available about the spatial position of other agents, it computes the energy relative to the choice.
- It accepts the position using a Montecarlo criterion:
  - If the new energy is lower than the current one, it accepts the move.
  - If the new energy is greater than the current one, it accepts the move with probability $e^{-\frac{E_{new}-E_{current}}{T}}$.

The CCP simulation program

- Currently, this approach is implemented in Prolog
- It uses the Linda package for communication (position are stored on the shared tuple space)
- The energy functions are computed by C functions imported in the Prolog code.
- The code is intrinsically parallel (but the bottleneck is communication).

Prolog–Linda code

```
simulation( S ) :-
      out(pos(1,initpos_1)), ..., out(pos(n,initpos_n)),
      out(trigger(1)),  ..., out(trigger(n)),
      amino(1, S ) ||  ... || amino(n, S).

amino(i,S) :-
      in(trigger(i)),
      get_positions([pos(1,Pos_1),..., pos(n,Pos_n)]),
      update_position(i,S,[pos(1,Pos_1),...,pos(n,Pos_n)], Newpos),
      out(pos(i,Newpos)),
      out(trigger(1)),..., out(trigger(i-1)),
      out(trigger(i+1)), ..., out(trigger(n)),
      amino(i,S).
```

Results

- With current parameters we are able to simulate the formation of secondary structures and of small proteins.
- We are studying a 4 level multiagent optimization architectures and, in particular, communication and cooperation strategies.
- We are replacing the Linda code with a shared memory C code.
- We wish to use this distributed multiagent model as a general minimization model.

END of PART 2

Other approaches

- Of course the problem is faced with various methods (see [Neumaier97] for a review for mathematicians)
- Those with best results are based on *homology* and *threading*.
- In `http://www.rcsb.org/pdb/` 29.500 structures are deposited (not all independent!)
- One can look for a *homologous* protein (small changes/removals/insertions between the primary structures).
- If it is found, its tertiary structure is selected and weakly re-arranged (threading phase) for the new protein.

Other approaches ... can use constraints

- If an homologous protein is not found, but various subsequences are found in the PDB,
- Each of those sequences is associated to a rigid local structure.
- The problem is reduced to the optimization of the energy assigning the positions to the unknown parts. This is similar of what is done using secondary structure constraints!
- Molecular Dynamics methods are precise but slow and can fall into local minima. They can start from a FCC Constraint-Based prediction!

Conclusions

- We have seen the definition of the Protein Structure Prediction Problem/Protein folding problem.
- We have focused on simplified models (of space and energy).
- We have seen two uses of Constraint (logic/concurrent) programming for attacking it:
  - As a constrained minimization problem (work with Alessandro Dal Palù and Federico Fogolari, and with Rolf Backofen,Enrico Pontelli, Sebastian Will, Sandro Bozzoli, Matteo Burato, and Fausto Spoto).
  - As (abstract) simulation (work with Luca Bortolussi, Federico Fogolari, and Alessandro Dal Palù)
- The constraint approaches can be integrated in existing tools.
- A natural and useful application of declarative programming.
- There is a lot of work to do.

Suggested Readings

- P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. John Wiley & Sons, 2001.

- A. Neumaier. Molecular modeling of proteins and mathematical prediction of protein structure. *SIAM Review*, 39:407–460, 1997.

- S. Miyazawa and R. L. Jernigan. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology*, 256(3):623–644, 1996.

- A. Kolinski and J. Skolnick. Reduced models of proteins and their applications. *Polymers* 45:511-524, 2004.

- T. Veitshans, D. Klimov, and D. Thirumalai. Protein folding kinetics: timescales, pathways and energy landscapes in terms of sequence-dependent properties. *Folding & Design*, 2:1–22, 1996.