Lattice Models: The Simplest Protein Model

The HP-Model (Lau & Dill, 1989)

- model only hydrophobic interaction
  - alphabet \( \{H, P\} \); \( H/P = \) hydrophobic/polar
  - energy function favors HH-contacts
- structures are discrete, simple, and originally 2D
  - model only backbone (C-\(\alpha\)) positions
  - structures are drawn (originally) on a square lattice \( \mathbb{Z}^2 \)
    without overlaps: Self-Avoiding Walk

Example
HP-Model Definition

Definition
The HP-model is a protein model, where

- Sequence $s \in \{H, P\}^n$
- Structure $\omega : [1..n] \rightarrow L$ (e.g. $L = \mathbb{Z}^2, L = \mathbb{Z}^3$), s.t.
  1. for all $1 \leq i < n$:
     \[ d(\omega(i), \omega(i + 1)) = d_{\text{min}}(L) \quad [d_{\text{min}}(\mathbb{Z}^2) = 1] \]
  2. for all $1 \leq i < j \leq n$ : $\omega(i) \neq \omega(j)$
- Energy function $E(s, \omega) = \sum_{1 \leq i < j \leq n} E_{s_i, s_j} \Delta(\omega(i), \omega(j))$

where $E = \begin{pmatrix} H & P \\ H & -1 & 0 \\ P & 0 & 0 \end{pmatrix}$

and $\Delta(p, q) = \begin{cases} 1 & \text{if } d(p, q) = d_{\text{min}}(L) \\ 0 & \text{otherwise} \end{cases}$
HP-Model Definition

Definition

The HP-model is a protein model, where

- Sequence $s \in \{H, P\}^n$
- Structure $\omega : [1..n] \rightarrow L$ (e.g. $L = \mathbb{Z}^2$, $L = \mathbb{Z}^3$), s.t.
  1. for all $1 \leq i < n$:
     $$d(\omega(i), \omega(i + 1)) = d_{\text{min}}(L) \quad [d_{\text{min}}(\mathbb{Z}^2) = 1]$$
  2. for all $1 \leq i < j \leq n$:
     $\omega(i) \neq \omega(j)$
- Energy function $E(s, \omega) = \sum_{1 \leq i < j \leq n} E_{s_i, s_j} \Delta(\omega(i), \omega(j))$,

where $E = \begin{pmatrix} H & P \\ H & -1 & 0 \\ P & 0 & 0 \end{pmatrix}$

and $\Delta(p, q) = \begin{cases} 1 & \text{if } d(p, q) = d_{\text{min}}(L) \\ 0 & \text{otherwise} \end{cases}$
Structures in the HP-Model

Sequence HPPHPH
How many structures are there?
Self-avoiding Walks of the Square Lattice (without Symmetry)

Naive enumeration not possible. Even NP-complete:

B. Berger, T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) Model is NP-complete. RECOMB’98

How many structures are there?
Self-avoiding Walks of the Square Lattice (without Symmetry)

Naive enumeration not possible. Even NP-complete:

B. Berger, T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) Model is NP-complete. RECOMB’98

Constraint Programming (CP)

- Model and solve *hard combinatorial problems* as CSP by search and propagation
- cf. ILP, but CP offers more flexible modeling and differs in solving strategies

**Definition**

A *Constraint Satisfaction Problems (CSP)* consists of

- variables $\mathcal{X} = \{X_1, \ldots, X_n\}$,
- the domain $D$ that associates finite domains $D_1 = D(X_1), \ldots, D_n = D(X_n)$ to $\mathcal{X}$.
- a set of constraints $C$.

A solution is an assignment of variables to values of their domains that satisfies the constraints.
## Commercial Impact of Constraints Programming

<table>
<thead>
<tr>
<th>Company</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michelin and Dassault, Renault</td>
<td>Production planning</td>
</tr>
<tr>
<td>Lufthansa, Swiss Air, ...</td>
<td>Staff planning</td>
</tr>
<tr>
<td>Nokia</td>
<td>Software configuration</td>
</tr>
<tr>
<td>Siemens</td>
<td>Circuit verification</td>
</tr>
<tr>
<td>French National Railway Company</td>
<td>Train schedule</td>
</tr>
</tbody>
</table>

...
CP Example: The N-Queens Problem

4-Queens: place 4 queens on $4 \times 4$ board without attacks
CP Example: The N-Queens Problem

4-Queens: place 4 queens on $4 \times 4$ board without attacks
CP Example: The N-Queens Problem

4-Queens: place 4 queens on $4 \times 4$ board without attacks
CP Example: The N-Queens Problem

4-Queens: place 4 queens on $4 \times 4$ board without attacks
4-Queens: place 4 queens on $4 \times 4$ board without attacks
Model 4-Queens as CSP (Constraint Model)

• Variables $X_1, \ldots, X_4$
  
  $X_i = j$ means “queen in column $i$, row $j$”

• Domains $D(X_i) = \{1, \ldots, 4\}$ for $i = 1..4$

• Constraints (for $i, i' = 1..4$ and $i \neq i'$)
  
  $X_i \neq X_{i'}$ (no horizontal attack)
  
  $i - X_i \neq i' - X_{i'}$ (no attack in first diagonal)
  
  $i + X_i \neq i' + X_{i'}$ (no attack in second diagonal)
Solving 4-Queens by Search and Propagation, $X_1 = 1$

\[
X_1, \ldots, X_4
\]

\[
D(X_i) = \{1, \ldots, 4\} \text{ for } i = 1..4
\]

\[
X_i \neq X_{i'}, \ i - X_i \neq i' - X_{i'}, \ i + X_i \neq i' + X_{i'}
\]
Solving 4-Queens by Search and Propagation, $X_1 = 1$

$X_1, \ldots, X_4$

$D(X_1) = \{1\}, D(X_i) = \{1, \ldots, 4\}$ for $i = 2..4$

$X_i \neq X_{i'}$, $i - X_i \neq i' - X_{i'}$, $i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 1$

$X_1, \ldots, X_4$

$D(X_1) = \{1\}, \ D(X_2) = \{3, 4\}, \ D(X_3) = \{2, 4\}, \ D(X_4) = \{2, 3\}$

$X_i \neq X_{i'}, \ i - X_i \neq i' - X_{i'}, \ i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 1$

$D(X_1) = \{1\}$, $D(X_2) = \{3, 4\}$, $D(X_3) = \{4\}$, $D(X_4) = \{2, 3\}$

$X_i \neq X_{i'}$, $i - X_i \neq i' - X_{i'}$, $i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 1$

$D(X_1) = \{1\}, D(X_2) = \{3, 4\}, D(X_3) = \{\}, D(X_4) = \{2, 3\}$

$X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 2$

\[ X_1, \ldots, X_4 \]

\[ D(X_i) = \{1, \ldots, 4\} \text{ for } i = 1..4 \]

\[ X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'} \]
Solving 4-Queens by Search and Propagation, $X_1 = 2$

$X_1, \ldots, X_4$

$D(X_1) = \{2\}, D(X_i) = \{1, \ldots, 4\}$ for $i = 2..4$

$X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 2$

$X_1, \ldots, X_4$

$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1, 3\}, D(X_4) = \{1, 3, 4\}$

$X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 2$

$X_1, \ldots, X_4$

$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1\}, D(X_4) = \{3, 4\}$

$X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'}$
Solving 4-Queens by Search and Propagation, $X_1 = 2$

$X_1, \ldots, X_4$

$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1\}, D(X_4) = \{3\}$

$X_i \neq X_{i'}, i - X_i \neq i' - X_{i'}, i + X_i \neq i' + X_{i'}$
Definition
A Constraint Optimization Problem (COP) is a CSP together with an objective function $f$ on solutions.
A solution of the COP is a solution of the CSP that maximizes/minimizes $f$.

Solving by Branch & Bound Search
Idea of B&B:
- Backtrack & Propagate as for solving the CSP
- Whenever a solution $s$ is found, add constraint “next solutions must be better than $f(s)$”.
The problem

**IN:** sequence $s$ in $\{H, P\}^n$

```
HHPPPHHPHPPHHPHPPHHPPPHPPPHPPHHH
```

**OUT:** self avoiding walk $\omega$ on cubic/fcc lattice with minimal HP-energy $E_{HP}(s, \omega)$
A First Constraint Model

• Variables $X_1, \ldots, X_n, Y_1, \ldots, Y_n, Z_1, \ldots, Z_n$ and $HHContacts$

$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$ is the position of the $i$th monomer $\omega(i)$

• Domains

$D(X_i) = D(Y_i) = D(Z_i) = \{-n, \ldots, n\}$

• Constraints

1. positions $i$ and $i + 1$ are neighbored (chain)
2. all positions differ (self-avoidance)
3. relate $HHContacts$ to $X_i, Y_i, Z_i$

$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
Solving the First Model

• Model is a COP (Constraint Optimization Problem)
• Branch and Bound Search for Minimizing Energy
• (Add Symmetry Breaking)
• How good is the propagation?
• Main problem of propagation: bounds on contacts/energy
  From a partial solution, the solver cannot estimate the maximally possible number of HH-contacts well.
Steps

1. Core Construction
2. Mapping
The Advanced Approach: Cubic & FCC

Steps

1. Bounds
2. Core Construction
3. Mapping
Computing Bounds

- Prepares the construction of cores
- How many contacts are possible for $n$ monomers, if freely distributed to lattice points
- Answering the question will give information for core construction
- Main idea: split lattice into layers consider contacts
  - within layers
  - between layers
Layers: Cubic & FCC Lattice
Layers: Cubic & FCC Lattice
Contacts

Contacts =
Layer contacts + Contacts between layers

• **Bound Layer contacts:** Contacts \( \leq 2 \cdot n - a - b \)

• **Bound Contacts between layers**
  - cubic: one neighbor in next layer
    \( Contacts \leq \min(n_1, n_2) \)
  - FCC: four neighbors in next layer
    \( i \) – points
Layer $L_1: n_1, a_1, b_1, m_{nc1}, m_{nt1}, m_{x1}$

Number of $i$-points $#i$ in $L_1$

$#4 = n_1 - a_1 - b_1 + 1 + m_{nc1}$
$#3 = m_{x1} - 2(m_{nc1} - m_{nt1})$
$#2 = 2a_1 + 2b_1 - 4 - 2#3 - 3m_{nc1} - m_{nt1}$
$#1 = #3 + 2m_{nc1} + 2m_{nt1} + 4$
Contacts between Layers

Layer $L_1 : n_1, a_1, b_1, m_{nc1}, m_{nt1}, m_{x1},$ Layer $L_2 : n_2$

Theorem (Number of contacts between layers)

(Eliminate parameter $m_{x1}$)

$$\#3' = \text{maximal number 3-points for } n_1, a_1, b_1, m_{nc1}, m_{nt1}$$

$$\iff \#2' = 2a_1 + 2b_1 - 4 - 2\#3' - 4m_{nc1}$$

$$\#1' = \#3' + 4m_{nc1} + 4 \quad \#4' = \#4$$

(Distribute $n'$ points optimally to $i$-points in $L_1$)

$$b_4 = \min(n_2, \#4') \quad b_3 = \min(n_2 - b_4, \#3')$$

$$b_2 = \min(n_2 - b_4 - b_3, \#2') \quad b_1 = \min(n_2 - b_4 - b_3 - b_2, \#1')$$

Contacts between $L_1$ and $L_2 \leq 4 \cdot b_4 + 3 \cdot b_3 + 2 \cdot b_2 + b_1$
Recursion Equation for Bounds

- \( B_C(n, n_1, a_1, b_1) \): Contacts of core with \( n \) elements and first layer \( L_1 : n_1, a_1, b_1 \)
- \( B_{\text{LC}}(n_1, a_1, b_1) \): Contacts in \( L_1 \)
- \( B_{\text{ILC}}(n_1, a_1, b_1, n_2, a_2, b_2) \): Contacts between \( E_1 \) and \( E_2 : n_2, a_2, b_2 \)
- \( B_C(n - n_1, n_2, a_2, b_2) \): Contacts in core with \( n - n_1 \) elements and first layer \( E_2 \)
Layer sequences

From Recursion:

- by Dynamic Programming: **Upper bound on number of contacts**
- by Traceback: **Set of layer sequences**

layer sequence = \((n_1, a_1, b_1), \ldots, (n_4, a_4, b_4)\)

Set of layer sequences gives distribution of points to layers in all point sets that possibly have maximal number of contacts
Core Construction

Problem

**IN:** number $n$, contacts $c$

**OUT:** all point sets of size $n$ with $c$ contacts

- Optimization problem
- Core construction is a hard combinatorial problem
Core construction: Modified Problem

**Problem**

**IN:** number \( n \), contacts \( c \), set of layer sequences \( S_{ls} \)

**OUT:** all point sets of size \( n \) with \( c \) contacts and layer sequences in \( S_{ls} \)

- Use constraints from layer sequences
- Model as constraint satisfaction problem (CSP)

\[(n_1, a_1, b_1), \ldots, (n_4, a_4, b_4)\]  
Core = Set of lattice points
Core Construction — Details

- Number of layers = length of layer sequence
- Number of layers in $x$, $y$, and $z$: Surrounding Cube
- enumerate layers $\Rightarrow$ fix cube $\Rightarrow$ enumerate points
Mapping Sequences to Cores

find structure such that

- H-Monomers on core positions $\rightarrow$ hydrophobic core
- all positions differ $\rightarrow$ self-avoiding
- chain connected $\rightarrow$ walk

compact core optimal structure
Mapping Sequence to Cores — CSP

Given: sequence $s$ of size $n$ and $n_H$ Hs

core $Core$ of size $n_H$

CSP Model

- **Variables** $X_1, \ldots, X_n$
  - $X_i$ is position of monomer $i$
- **Encode positions as integers**

\[
\begin{pmatrix}
  x \\
  y \\
  z
\end{pmatrix} \equiv M^2 \times x + M \times y + z
\]

(Unique encoding for 'large enough' $M$)

- **Constraints**
  1. $X_i \in Core$ for all $s_i = H$
  2. $X_i$ and $X_{i+1}$ are neighbors
  3. $X_1, \ldots, X_n$ are all different
Constraints for Self-avoiding Walks

- Single Constraints "self-avoiding" and "walk" weaker than their combination
- No efficient algorithm for consistency of combined constraint "self-avoiding walk"
- Relaxed combination: stronger and more efficient propagation

$k$-avoiding walk constraint

Example: 4-avoiding, but not 5-avoiding
Putting it together

Predict optimal structures by combining the three steps

1. Bounds
2. Core Construction
3. Mapping

Some Remarks

- Pre-compute optimal cores for relevant core sizes
  Given a sequence, only perform Mapping step
- Mapping to cores may fail!
  We use suboptimal cores and iterate mapping.
- Approach extensible to HPNX
  HPNX-optimal structures at least nearly optimal for HP.
Time efficiency

Prediction of one optimal structure
(“Harvard Sequences”, length 48 [Yue et al., 1995])

<table>
<thead>
<tr>
<th></th>
<th>CPSP</th>
<th>PERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,1 s</td>
<td>6,9 min</td>
<td></td>
</tr>
<tr>
<td>0,1 s</td>
<td>40,5 min</td>
<td></td>
</tr>
<tr>
<td>4,5 s</td>
<td>100,2 min</td>
<td></td>
</tr>
<tr>
<td>7,3 s</td>
<td>284,0 min</td>
<td></td>
</tr>
<tr>
<td>1,8 s</td>
<td>74,7 min</td>
<td></td>
</tr>
<tr>
<td>1,7 s</td>
<td>59,2 min</td>
<td></td>
</tr>
<tr>
<td>12,1 s</td>
<td>144,7 min</td>
<td></td>
</tr>
<tr>
<td>1,5 s</td>
<td>26,6 min</td>
<td></td>
</tr>
<tr>
<td>0,3 s</td>
<td>1420,0 min</td>
<td></td>
</tr>
<tr>
<td>0,1 s</td>
<td>18,3 min</td>
<td></td>
</tr>
</tbody>
</table>

- **CPSP**: “our approach”, constraint-based
- **PERM** [Bastolla et al., 1998]: stochastic optimization
Many Optimal Structures

Sequence HPPHPPPHP

• There can be many ...
• HP-model is degenerated
• Number of optimal structures $=$ degeneracy
Completeness

Predicted number of all optimal structures (“Harvard Sequences”)

<table>
<thead>
<tr>
<th>CPSP</th>
<th>CHCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.677.113</td>
<td>1500 \times 10^3</td>
</tr>
<tr>
<td>28.180</td>
<td>14 \times 10^3</td>
</tr>
<tr>
<td>5.090</td>
<td>5 \times 10^3</td>
</tr>
<tr>
<td>1.954.172</td>
<td>54 \times 10^3</td>
</tr>
<tr>
<td>1.868.150</td>
<td>52 \times 10^3</td>
</tr>
<tr>
<td>106.582</td>
<td>59 \times 10^3</td>
</tr>
<tr>
<td>15.926.554</td>
<td>306 \times 10^3</td>
</tr>
<tr>
<td>2.614</td>
<td>1 \times 10^3</td>
</tr>
<tr>
<td>580.751</td>
<td>188 \times 10^3</td>
</tr>
</tbody>
</table>

- **CPSP**: “our approach”
- **CHCC** [Yue et al., 1995]: complete search with hydrophobic cores
Unique Folder

- HP-model degenerated
- Low degeneracy $\approx$ stable $\approx$ protein-like
- Are there protein-like, unique folder in 3D HP models?
- How to find out?
Unique Folder

- HP-model degenerated
- Low degeneracy $\approx$ stable $\approx$ protein-like
- Are there protein-like, unique folder in 3D HP models?
- How to find out?

MC-search through sequence space
Unique Folder

- HP-model degenerated
- Low degeneracy $\approx$ stable $\approx$ protein-like
- Are there protein-like, unique folder in 3D HP models?
- How to find out?

Yes: many, e.g. about 10,000 for $n=27$
Software: CPSP Tools

http://cpsp.informatik.uni-freiburg.de:8080/index.jsp

CPSP Tools

Constraint-based Protein Structure Prediction

Bioinformatics Group
Albert-Ludwigs-University Freiburg

web-tools version 1.1.1 (06.04.2011)

The CPSP-tools package provides programs to solve exactly and completely the problems typical of studies using 3D lattice protein models. Among the tasks addressed are the prediction of globally optimal and/or suboptimal structures as well as sequence design and neutral network exploration.

Choose a tool from the left for ad hoc usage
(CPSP-tools version 2.4.2) (LatPack version 1.7.2)

or

Download the full CPSP-tools or LatPack package for local usage!

If you use the CPSP-tools please cite the following publications:

- Martin Mann, Sebastian Will, and Rolf Backofen.
  CPSP-tools - Exact and Complete Algorithms for High-throughput 3D Lattice Protein Studies.