Solving large protein design problems modeled as cost function networks
Guaranteed Discrete Energy Optimization on Large Protein Design Problems.
*Journal of chemical theory and computation.*

David Simoncini
D. Allouche, S. de Givry,
C. Delmas, S. Barbe (INSA), T. Schiex

May 2016 - The MIAT seminars
What is a protein?

(Thank you wikipedia)

Amino acids, proteins

- Proteins are linear chains of amino-acids (20 natural AAs).
- All AAs share a common “core” and have a variable side-chain.

Side-chains are flexible (ARG)
Protein Design

Why?

- Proteins have various functions in the cell: catalysis, signaling, recognition, regulation...
- Efficient, biodegradable, $10^6$ to $10^{20}$ speedups
- Nano-technologies (shape more than function).
- Medicine, cosmetics, food, bio-energies...
Protein Design

Protein function linked to its 3D shape through its amino acid composition.

Protein design’s aim
Identify sequences that have a suitable function (shape).

Issue
There are $20^n$ proteins of length $n$. Impossible to synthesize and test all of them.
Successes of Protein Design

- **1997**
  - Zinc-finger
    Dahiyat et al (Science 1997)

- **1998**
  - Coiled-coils
    Harbury et al (Science 1998)

- **2003**
  - A novel topology
    (Top7) Khulan et al (Science 2003)
  - Sensors & receptors toward non native ligands

- **2008**
  - New catalyst for a multi-step-reaction (retro-aldolase)
    Jiang et al (Science 2008)
  - De novo design of new functional enzyme (Kemp elimination)
    Rothlisberger et al (Nature 2008)

- **2010**
  - Design of Biocatalyst for a stereoselective & bimolecular reaction (Diels-alder)
    Siegel et al (Science 2010)
The CPD problem

Rigid backbone variant

1. Assume a rigid protein backbone.
2. Choose 1 AA among possible ones at each mutable position.

Search Space

Fully discrete description, defined by a choice of rotamer (AA $\times$ conformation) for each position.

Pairwise decomposable energy function

\[ E(c) = E_\emptyset + \sum_{i=1}^{n} E(i_r) + \sum_{i<j} E(i_r, j_s) \]
Common approaches to CPD

DEE/A*

*Dead End Elimination:*

- Removes from the search space rotamers which are dominated.
- Can possibly remove close to optimal solutions.

*A* algorithm:

- Best-first search tree-based algorithm.
- A heuristic gives a lower bound on the cost of each path in the tree.

Meta-heuristics

- *Monte-Carlo Simulated Annealing* (Rosetta).
- ...
What is a Graphical Model?

Informal

1. A set of discrete variables, each with a domain
2. We want to define a joint function (energy) 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 (energy) 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. Allows to describe knowledge on a lot of variables concisely
3. Usually hard to manipulate (NP-hard queries).
A Cost Function Network is a Graphical Model

Cost Function 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\(^3\) (other: valued CSP\(^{14} \)).

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

A solution has cost \(< k\). Optimal if minimum cost.
Wooo beautiful artwork
Fixed BB discrete rotamers GMEC as a CFN

Straightforward

- Variables: mutable, flexible residues and rotamers
- Domains: available rotamers
- Cost functions:

\[ E(c) = E_\emptyset + \sum_{i=1}^{n} E(i_r) + \sum_{i<j} E(i_r, j_s) \]

Just shift all energies to make them non-negative.

Finding the GMEC is NP-hard\textsuperscript{12}
Our secret recipe

Four main ingredients

1. Depth First Branch and Bound
2. Good initial upperbound
3. Local consistency filtering induced lower bounds instead of DEE
4. Treewidth based problem decomposition
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$?
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
4. Else choose a residue $x_i$
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
4. Else choose a residue $x_i$
5. Split its domain in subsets
Depth First Branch and Bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. Is it $\geq k$?
3. If yes, backtrack
4. Else choose a residue $x_i$
5. Split its domain in subsets
6. For each subset

![Diagram of Depth First Branch and Bound](image)
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
4. Else choose a residue $x_i$
5. Split its domain in subsets
6. For each subset
   6.1 Recurse
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
4. Else choose a residue $x_i$
5. Split its domain in subsets
6. For each subset
   6.1 Recurse

1. When a solution is found, update $k$ to its energy.
2. DFS vs. $A^*$ (BFS): polynomial space vs. exponential space.
Depth First Branch and bound

Initial upper bound $k$

1. Compute a lower bound on the GMEC energy
2. is it $\geq k$ ?
3. If yes backtrack
4. Else choose a residue $x_i$
5. Split its domain in subsets
6. For each subset
   6.1 Recurse

1. When a solution is found, update $k$ to its energy.
2. DFS vs. $A^*$ (BFS): polynomial space vs. exponential space.
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0, k = 4$

Shift 1 to right $a$

Shift 1 to left $b$

Shift 1 from right $a$

Shift 1 from left $b$

Shift 1 from $x_1$ to $c_{\emptyset}$

$c_{\emptyset} = 1$

Preserves global energy below $k$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$

\[
\begin{array}{ccc}
X_1 & & X_2 \\
\text{a} & 1 & \text{a} \\
\text{b} & \text{b} & \\
X_1 & & X_2 \\
\text{1} & & \text{4} \\
\end{array}
\]

\begin{itemize}
\item Shift 1 from left \text{a}
\item Shift 1 from right \text{a}
\item Shift 1 from $X_1$ to $c_{\emptyset} = 1$
\item Preserves global energy below $k$
\end{itemize}
Improving the bound $c_∅$

Assume that initially $c_∅ = 0$, $k = 4$

Shift 1 to left $b$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$

Shift 1 to left $b$

Shift 1 from left $b$
Improving the bound $c_\emptyset$

Assume that initially $c_\emptyset = 0, k = 4$

Shift 1 to left $b$

\[ \begin{array}{c}
\text{x}_1 \\
1 \\
1 \\
\end{array} \quad \begin{array}{c}
\text{x}_2 \\
3 \\
\end{array} \quad \begin{array}{c}
\text{x}_1 \\
a \\
a \\
\end{array} \quad \begin{array}{c}
\text{x}_2 \\
a \\
b \\
\end{array} \]

⇓ Shift 1 from $\text{x}_1$ to $c_\emptyset$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$

Shift 1 to left $b$

\[
\begin{array}{cccc}
X_1 & X_2 & \leftarrow & \\
3 & & & 1 \\
\end{array}
\]

Shift 1 from left $b$

\[
\begin{array}{cccc}
X_1 & X_2 & \leftarrow & \\
1 & a & 3 & a \\
& b & & b \\
\end{array}
\]

\[
\downarrow \quad \text{Shift 1 from } x_1 \text{ to } c_{\emptyset}
\]

$c_{\emptyset} = 1$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$

Shift 1 to left $b$

\[
\begin{array}{ccc}
\text{x}_1 & \rightarrow & \text{x}_2 \\
 & & \leftarrow \\
\end{array}
\]

↓  Shift 1 from $x_1$ to $c_{\emptyset}$

$c_{\emptyset} = 1$

Preserves global energy below $k$
Improving the bound $c_{\emptyset}$

Assume that initially $c_{\emptyset} = 0$, $k = 4$.

Shift 1 to left $b$

\[
\begin{array}{c}
\text{x}_1 \\
\text{x}_2 \\
\end{array}
\quad
\begin{array}{c}
3 \\
\end{array}
\quad
\begin{array}{c}
\text{a} \\
1 \\
\text{b} \\
\end{array}
\quad
\begin{array}{c}
\text{x}_1 \\
\text{x}_2 \\
\end{array}
\quad
\begin{array}{c}
3 \\
a \\
b \\
\end{array}
\]

$\downarrow$

Shift 1 from $x_1$ to $c_{\emptyset}$

$c_{\emptyset} = 1$

Preserves global energy below $k$

DEE does not
Local consistencies

Optimize transformations to maximize lower bound

1. Arc Consistency$^{13}$
2. (Full) Directional Arc Consistency$^{9}$
3. Full Existential Directional Arc Consistency$^{10}$
4. Virtual Arc Consistency$^{4,5}$
5. Optimal Soft Arc Consistency (LP)$^{2,5}$
Tree decomposition

Tree of bags

Decomposition of a problem in a well-formed (RIP) tree of bags of variables.
Full redesign of 107 short proteins

Why full redesigns

1. Challenging
2. Used on $\beta_1$ domain of protein G to tune energy function parameters$^1$.

The designs

1. Structures extracted from the PDB (September 2014)
2. Length from 50 to 100 AA
3. Resolution better than 2 Å
4. Only representants at 30% identity
5. Talaris14 and Dunbrack’s 2010 rotamers
6. PyRosetta: relax + energy matrices
Looking for the Global Minimum Energy Configuration

How

1. Intel Xeon E5-2690 2.9GHz (Q1-2012 CPU)
2. Best of 1000 runs of fixbb Rosetta protocol (Simulated Annealing)
3. toulbar2: 100 hours limit.

La patate douce

https://bitbucket.org/satsumaimo/ptcfopd
Looking for the GMEC
Looking for the GMEC

toulbar2 (CFN)

1. 98 problems solved to optimality
2. Largest problem solved: $10^{234}$, 1.7 GB for energy matrix.
3. Smallest unsolved: $10^{206}$. 
Looking for the GMEC

toulbar2 (CFN)

1. 98 problems solved to optimality
2. Largest problem solved: \( 10^{234} \), 1.7 GB for energy matrix.
3. Smallest unsolved: \( 10^{206} \).

Rosetta/fixbb

1. Rosetta fixbb found the GMEC on 13 of these problems
2. These 13 problems took 90 hours for fixbb.
3. toulbar2 solved them to optimality in 36 hours.
Exploring Sequence/conformations around of the GMEC

All sequence/conformations in a 0.2 Rosetta unit threshold

1. Same 100h limit
Exploring Sequence/conformations around of the GMEC

All sequence/conformations in a 0.2 Rosetta unit threshold

1. Same 100h limit
2. Exhausted sequence/conformation space on 92/98 designs.
Exploring Sequence/conformations around of the GMEC

All sequence/conformations in a 0.2 Rosetta unit threshold

1. Same 100h limit
2. Exhausted sequence/conformation space on 92/98 designs.
3. Very fast sampling, but huge spaces (up to $1.42 \times 10^9$)
Diversity of situations

1. No clear tendency for simulated annealing success/failure pattern.
2. Not enough successes to see a trend?
Exploring sequences around the GMEC

Faster exploration of sequences only$^{15}$

- New “SCP branching” algorithm that explores the sequence space
- Allows to explore far larger energy gaps.
- Gives just one (sub)optimal conformation per sequence.

Faster exploration of sequences only$^{15}$

- A number of CFN algorithms injected directly in OSPREY$^6$
- Benefits to continuous/flexible BB design through DEEP$^8$, LUTE$^7$. 
Rosetta fixbb protocol: gap to optimality

- Blue: best over 1000 runs
Rosetta fixbb protocol: gap to optimality

- Blue: best over 1000 runs
- Red: all runs on all designs (worse may be off by 45 RU).
Distance to optimum as a function of space size

- Blue: best over 1,000 runs
Distance to optimum as a function of space size

- Blue: best over 1000 runs
- Red: average over 1000 runs.
Reliability, distance to optimum and size

- Blue: probability of finding the GMEC (sorted)

Histogram: # of unique sequences (x RU gap) (red: lower bound)

Protein design problems

Frequency GMEC found

Energy gap

0
0.075
0.15
0.225
0.3
2
4
6
8
Protein design problems

Frequency GMEC found

Energy gap
Reliability, distance to optimum and size

- Blue: probability of finding the GMEC (sorted)
- Red: energy gap to GMEC (sorted)
Reliability, distance to optimum and size

- Blue: probability of finding the GMEC (sorted)
- Red: energy gap to GMEC (sorted)
- Histogram: # of unique sequences ($\times RU \text{ gap}$) (red: lower bound)
What about sequences: Hamming dist. to GMEC

- Blue: best energy (2.4% core, 7% boundary, 10% surface).
What about sequences: Hamming dist. to GMEC

- Blue: best energy (2.4% core, 7% boundary, 10% surface).
- Red: average over 1 000 runs.
Distance to native as we get closer to the GMEC

Native sequence used to tune energy$^1,^{11}$. 
Distance to native as we get closer to the GMEC

Native sequence used to tune energy\(^1,11\).

<table>
<thead>
<tr>
<th>Type</th>
<th>native</th>
<th>fixbb best</th>
<th>GMEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charged</td>
<td>1,795</td>
<td>↑</td>
<td>1,996</td>
</tr>
<tr>
<td>Aromatic</td>
<td>585</td>
<td>↑</td>
<td>616</td>
</tr>
<tr>
<td>Polar</td>
<td>1,817</td>
<td>↓</td>
<td>1,730</td>
</tr>
<tr>
<td>Hydrophobic</td>
<td>2,585</td>
<td>↓</td>
<td>2,440</td>
</tr>
</tbody>
</table>

Cysteines in disulfide bridges: not counted.
Possible lessons

Monte Carlo sampling

- Fixbb SA becomes quickly unable to reach lowest energy regions
- Energy gap increases quickly with the number of mutable residues

Guarantees

- GMEC may be not crucial, but an upper bound on error is important
- Guaranteed optimum have different composition
- Talaris favorable for guaranteed optimization (but exponential barrier)
- Exhaustive enumeration can be very fast (but exponential size output)
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
2. Conformational entropy contribution to affinity
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
2. Conformational entropy contribution to affinity
3. Improve the “CPD” model
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
2. Conformational entropy contribution to affinity
3. Improve the “CPD” model
4. beyond pairwise decomposition$^7$
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
2. Conformational entropy contribution to affinity
3. Improve the “CPD” model
4. beyond pairwise decomposition
5. multistate (positive/negative)
Continue to contribute to CPD

1. Injected in OSPREY, contributes to “flexible” modeling
2. Conformational entropy contribution to affinity
3. Improve the “CPD” model
4. beyond pairwise decomposition
5. multistate (positive/negative)
6. symmetric and fragment design


References II


Seydou Traoré et al. “Fast search algorithms for computational protein design”. In: Journal of computational chemistry (2016).