ROTAMER OPTIMIZATION FOR PROTEIN DESIGN THROUGH MAP ESTIMATION AND PROBLEM-SIZE REDUCTION

Hong, Lippow, Tidor, Lozano-Perez. JCC. 2008.
Presented by Kyle Roberts

Problem Statement

- Protein structure prediction
  - Homology modeling
  - Side-chain placement
- Protein design problems
  - Given backbone and energy function find minimum energy side-chain conformation
- The **G**lobal **M**inimum **E**nergy **C**onformation (GMEC) problem

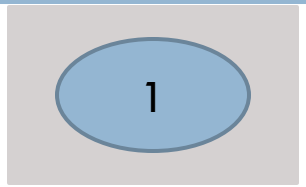
Current Approaches

- Dead-End Elimination (DEE)
- Branch-and-bound method (Leach, Lemon. *Proteins* 1998)
- Linear Programming
- Dynamic Programming
- Approximate Methods (SCMF, MC, BP)

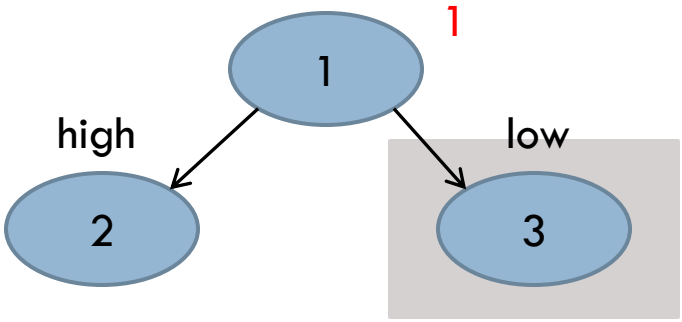
New Approach: BroMAP

- Branch-and-bound method with new subproblem-pruning method
- Focus on dense networks where all residues interact with one another
- Attack smaller sub-problems separately
- Can utilize DEE during sub-problems

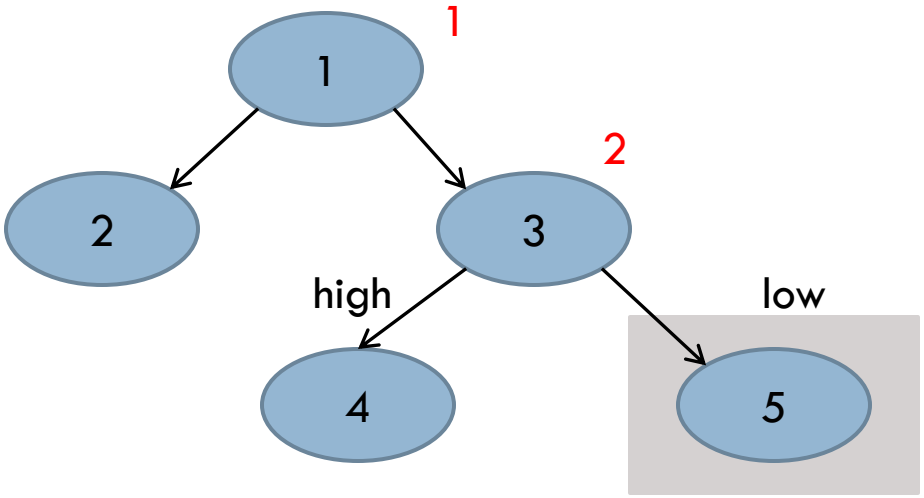
General Idea



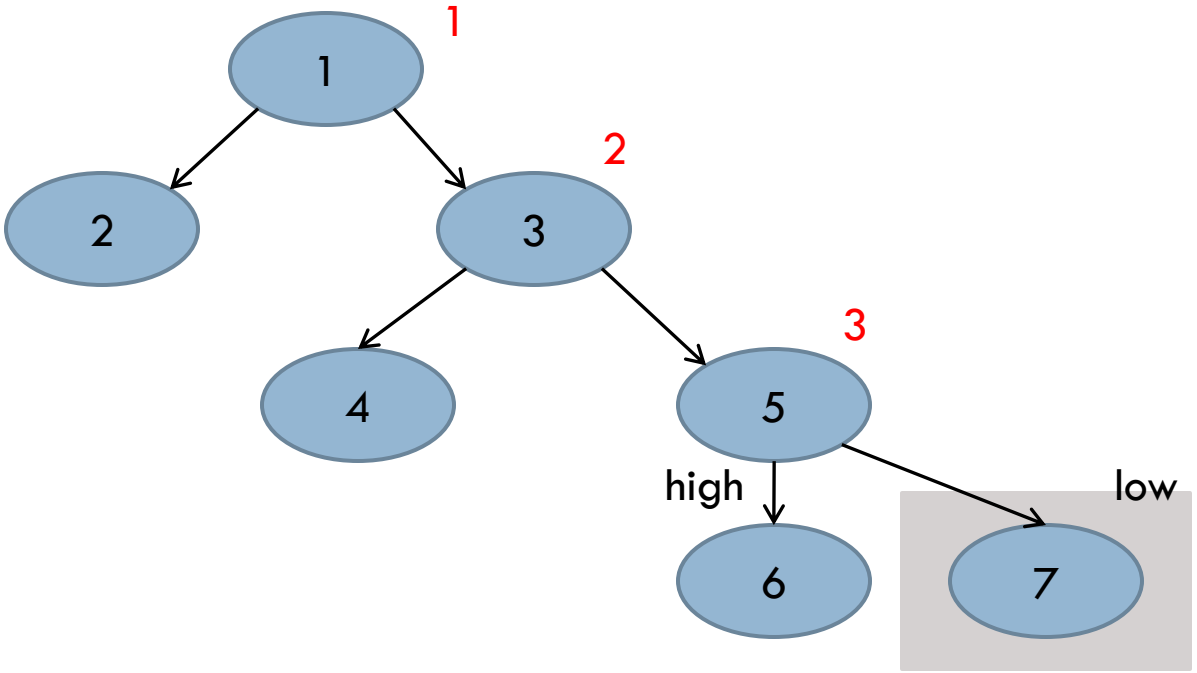
General Idea



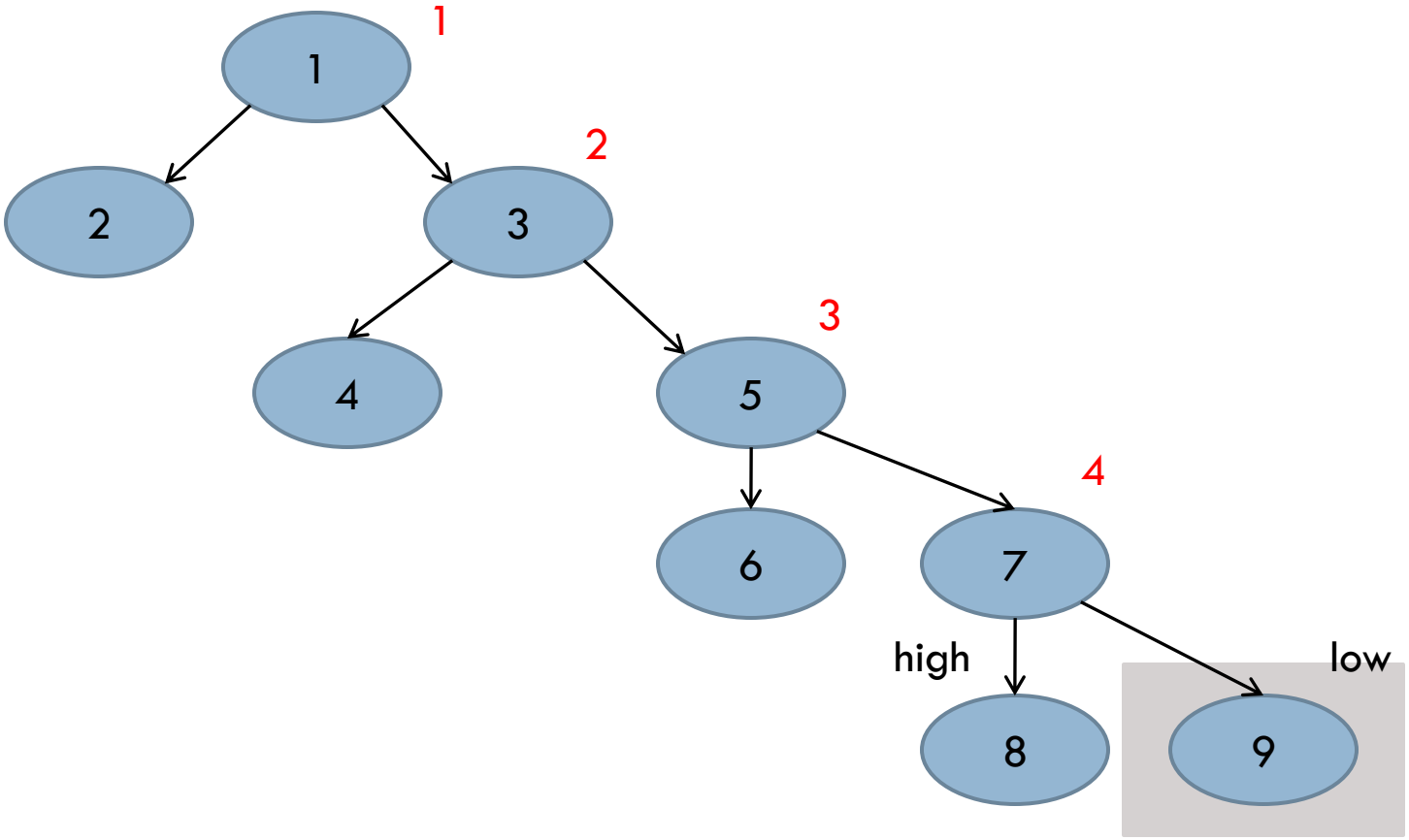
General Idea



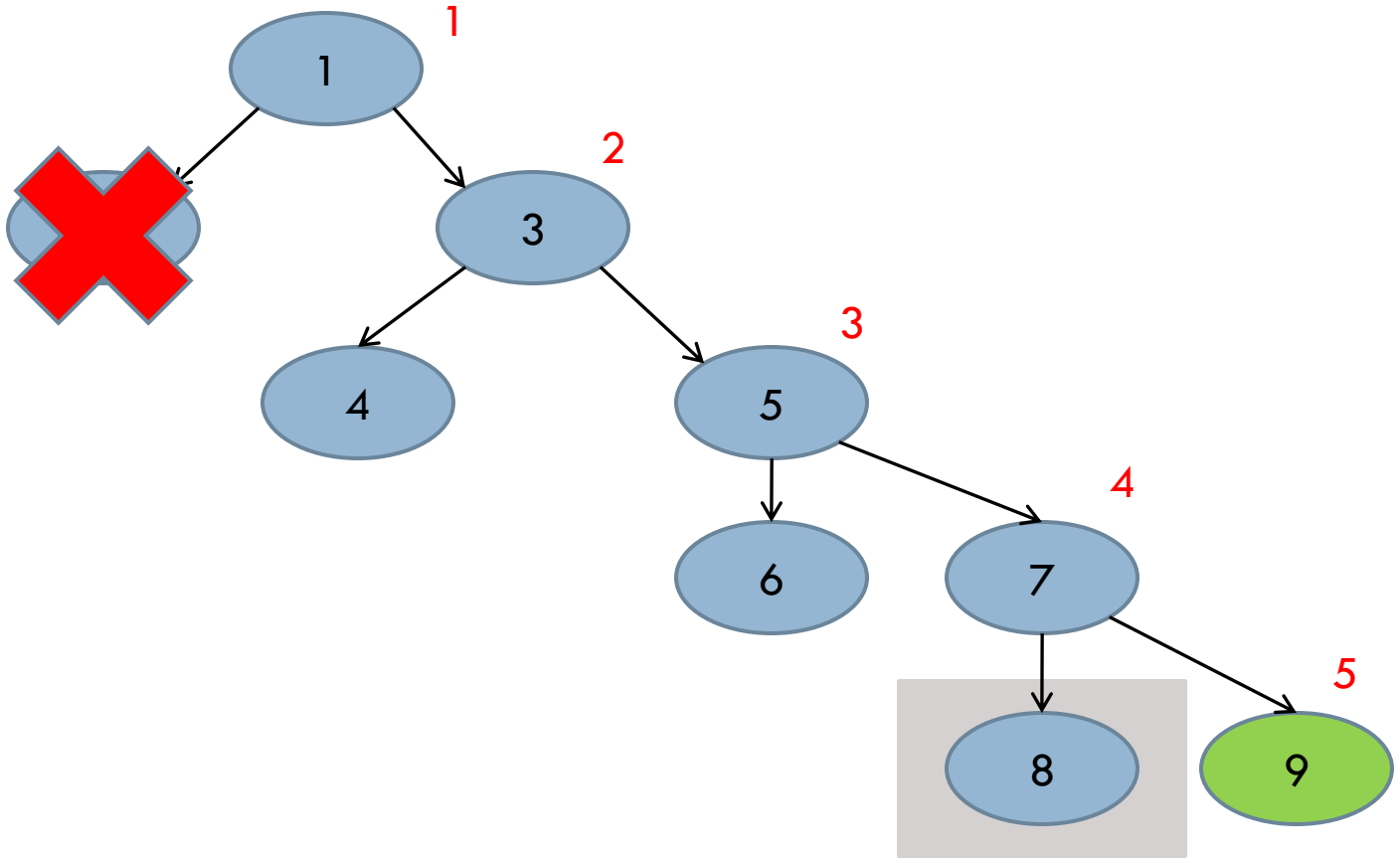
General Idea



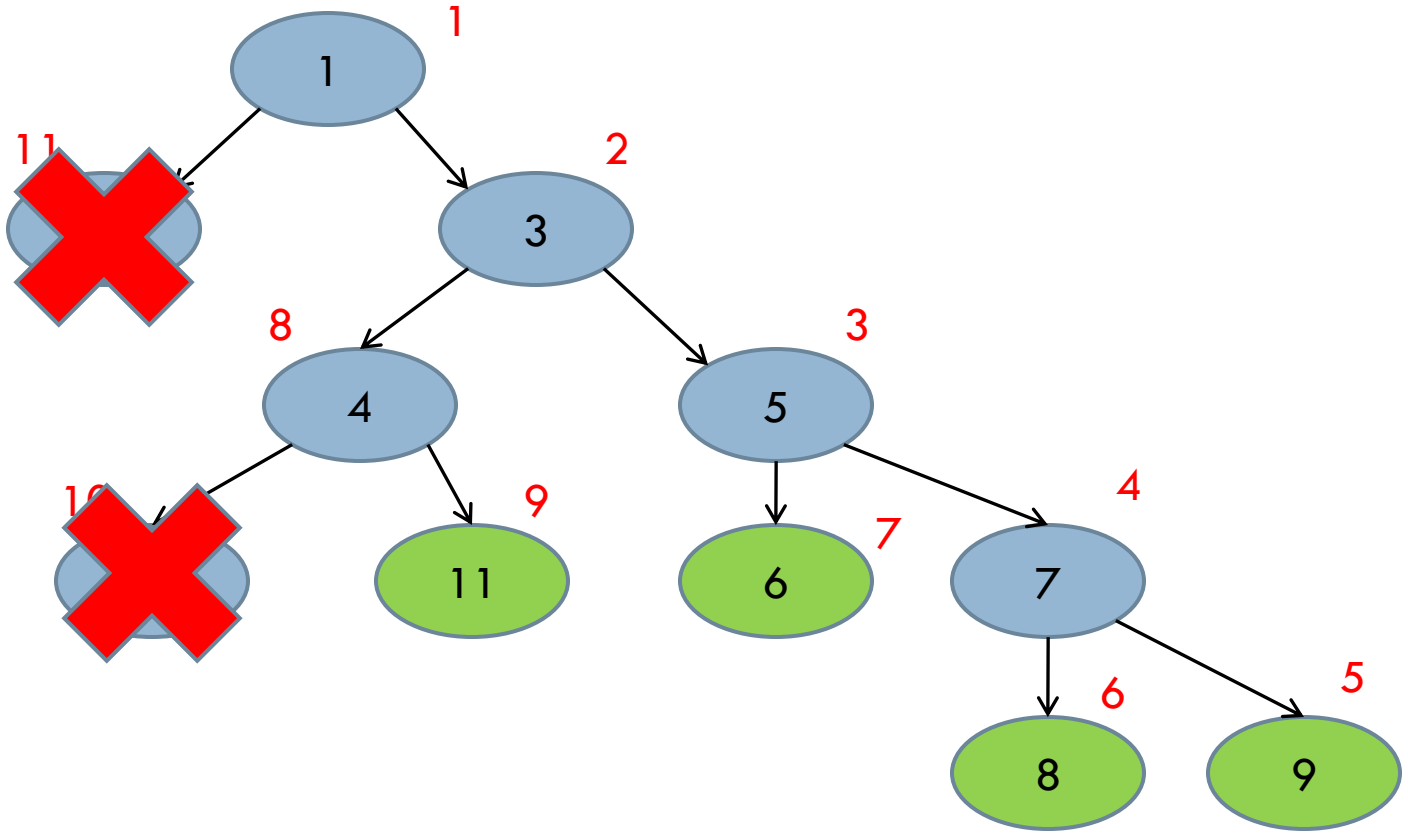
General Idea



General Idea



General Idea

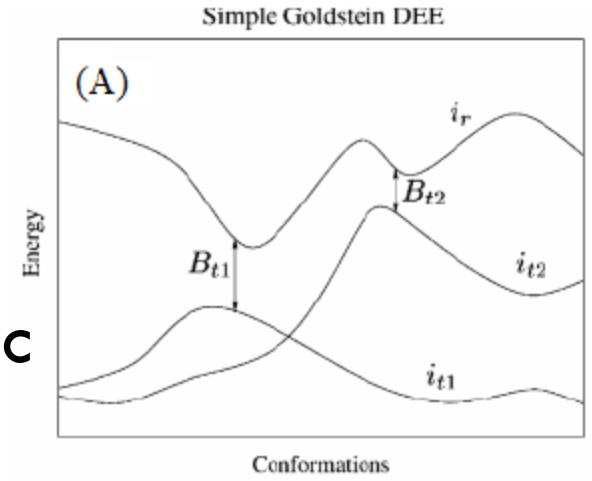


Recursive Steps

1. Select subproblem from queue
2. If easily solved, solve the subproblem
    1. Update the minimum energy (U) seen and return
3. Compute lower bound (LB) and upper bound (UB) on minimum energy for subproblem
    1. If UB is less than U, set U to the UB
4. Prune subproblem if LB is greater than U
5. Exclude ineligible conformations from search (DEE)
6. Pick one residue, and split rotamers into two groups
7. Add child subproblems to the queue and return

2. Solving Subproblems

- Use DEE/A* to solve subproblems directly
- Goldstein singles
- Singles using split flags
- Logical singles-pairs elimination
- Goldstein's condition with one magic bullet
- Logical singles-pairs elimination
- Do unification if possible
- (Small enough: <200,000 rotamers)



N.A. Pierce, J.A. Spriet, J. Desmet, S.L. Mayo. JCC. 2000.

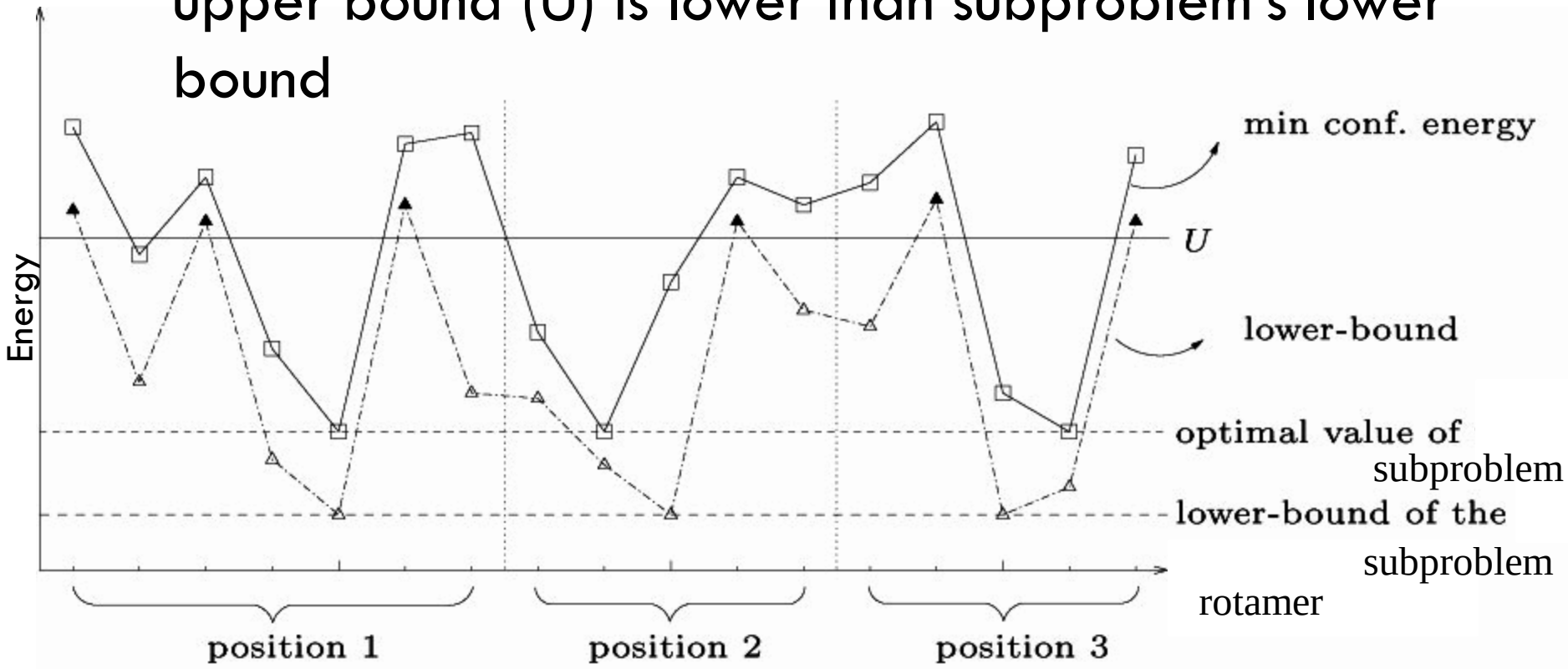
3. Bounding Subproblems

- Tree-reweighted max-product algorithm (TRMP)
- Relatively low computation cost
- Can be used to compute lower-bounds for parts of the conformational space efficiently

- (Discussed later)

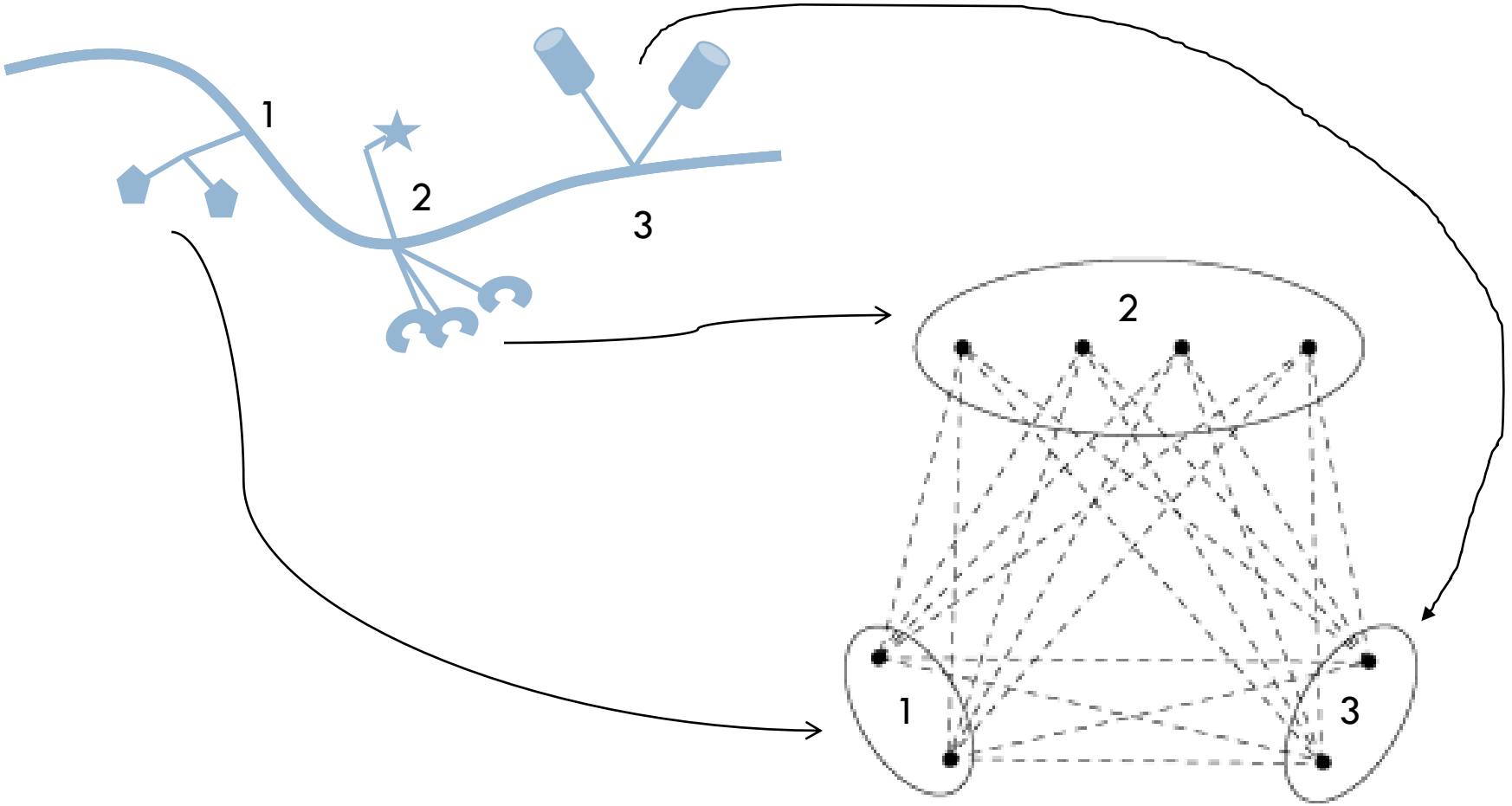
4/5. Prune Subproblem and Rotamers

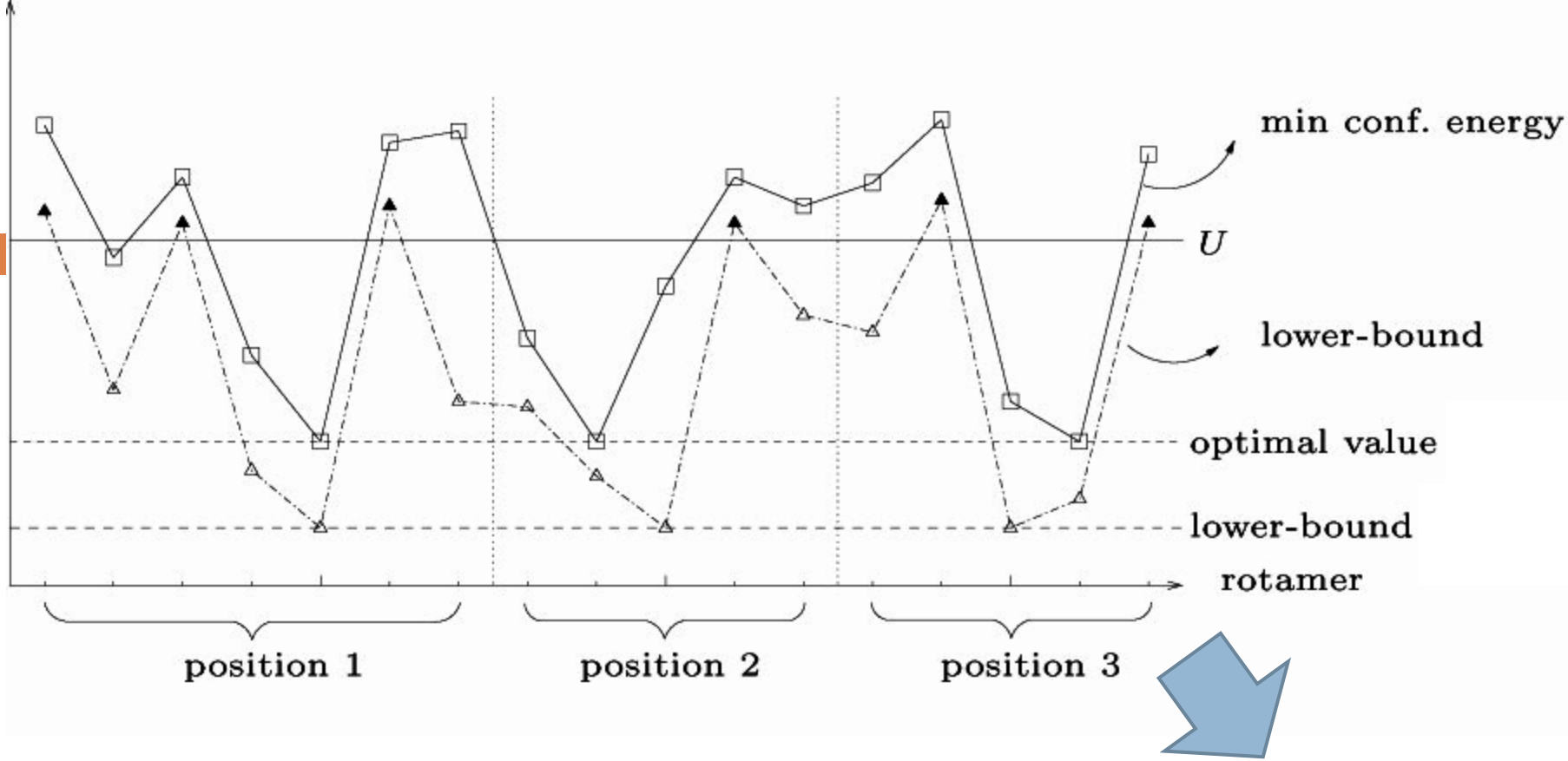
- Subproblem can be pruned if the current global upper bound (U) is lower than subproblem's lower bound

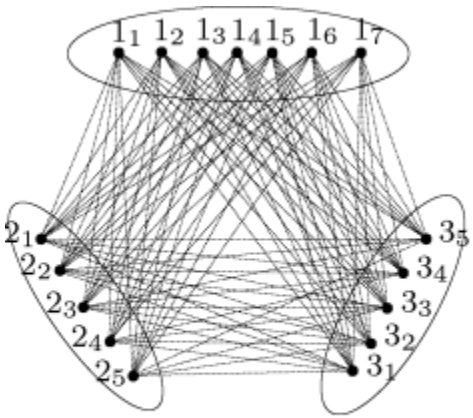


Figures taken from: Hong, Lippow, Tidor, Lozano-Pérez. JCC. 2008 unless otherwise stated

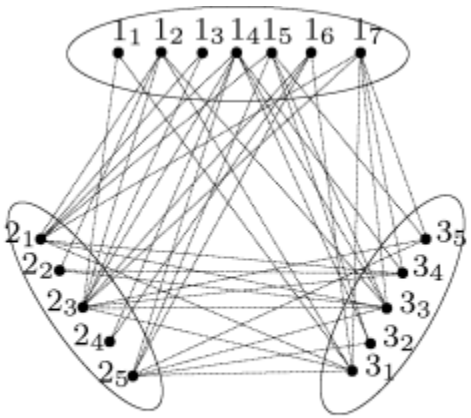
Representing Problem as Graph



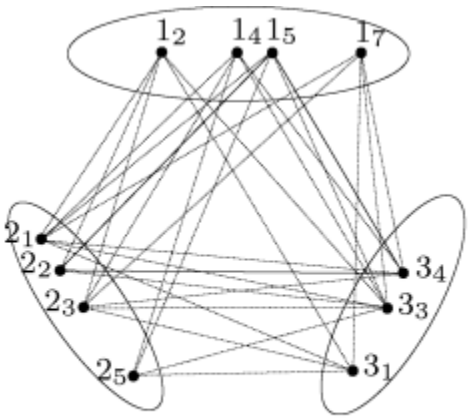




(a) Original subproblem.



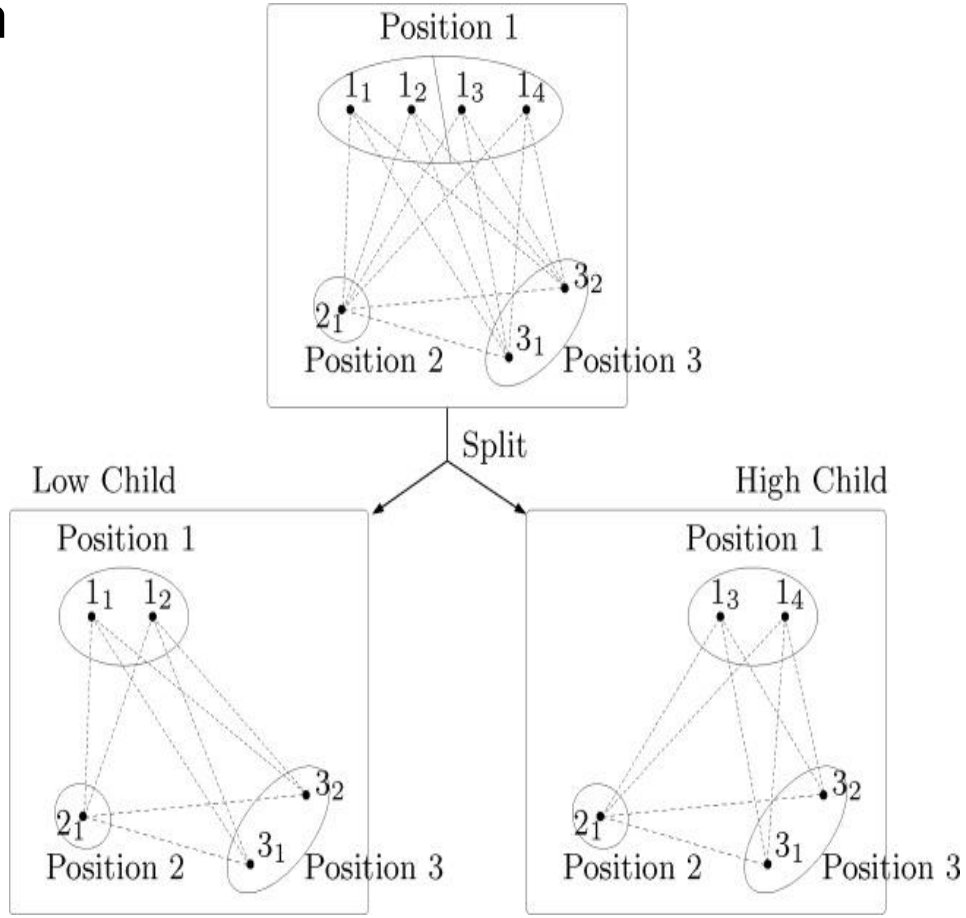
(b) Rotamer-pair elimination.



(c) Rotamer elimination.

Subproblem Splitting

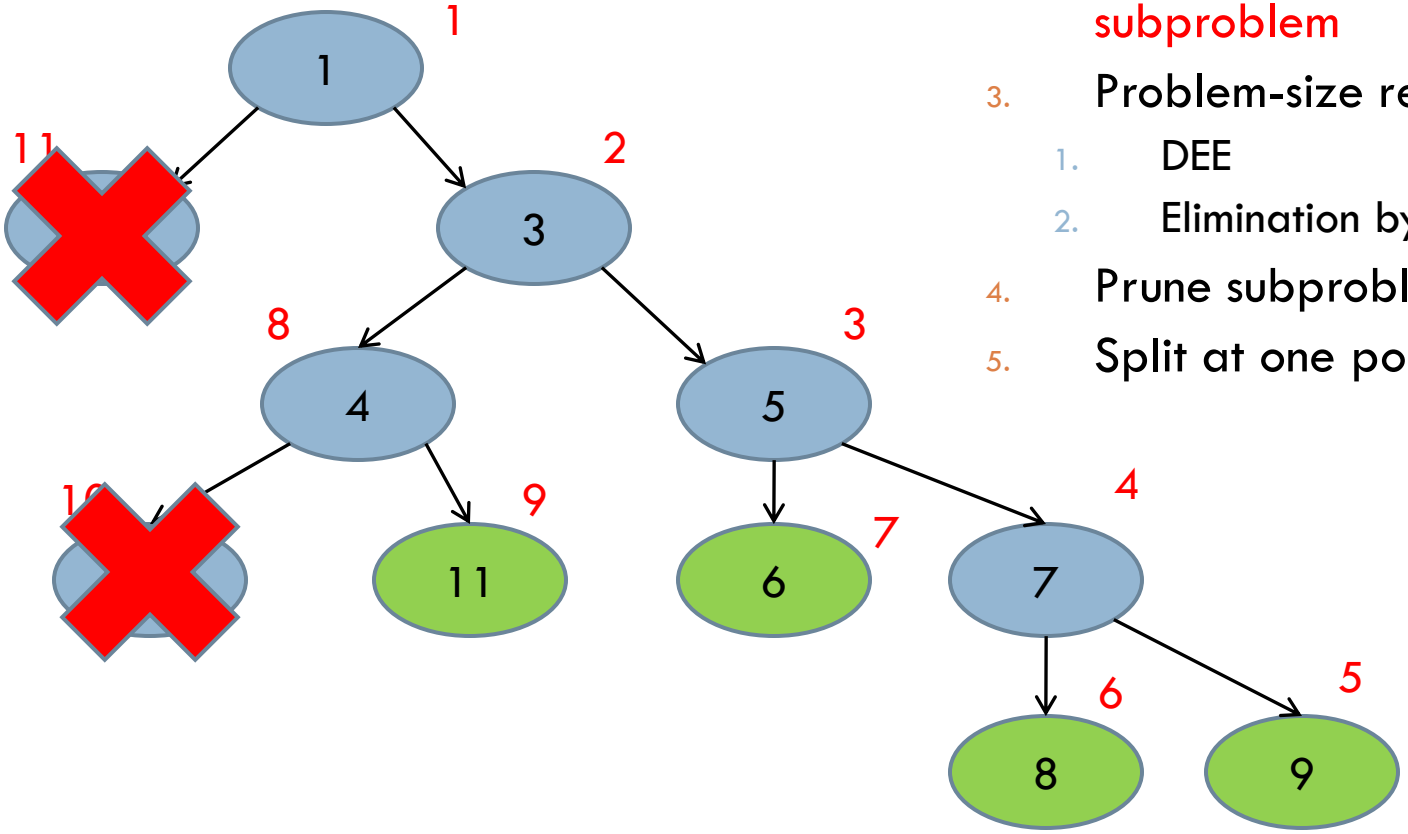
- Split rotamers at a given position into two groups (high lower bounds and low lower bounds)
- Splitting position is selected to so that maximum and minimum rotamer lower bounds is large



Subproblem Selection

- Use depth first search to choose which subproblem to expand
- This leads to quickly finding a good upper bound in order to allow additional pruning

Summary



1. Direct solution by DEE
2. Lower/Upper bound subproblem
3. Problem-size reduction:
    1. DEE
    2. Elimination by TRMP bounds
4. Prune subproblem if possible
5. Split at one position

MAP Estimation

- Maximum a-posteriori (MAP) estimation problem:
- Find a MAP assignment $\mathbf{x}^*$ such that

$$\mathbf{x}^* \in \arg\max_{\mathbf{x} \in \chi} p(\mathbf{x})$$

- We can convert this to the GMEC problem if

$$p(\mathbf{x}) = \frac{1}{Z}\exp[-e(\mathbf{x})] \qquad \text{where } e(\mathbf{x}) = \text{Energy of conformation } \mathbf{x}$$

$|\mathbf{x}| =$ number of residue positions

$$\chi = R_1 \times R_2 \times \ldots \times R_n$$

- Maximizing the probability => minimizing energy

Max Marginals find MAP Estimation

- The max marginal, μ_i, is defined as the maximum of $p(\mathbf{x})$ when one position x_i is constrained to a given rotamer: $\mu_i(x_i) = \kappa_i \max_{\{x' | x'_i = x_i\}} p(\mathbf{x}')$

$$\mu_{ij}(x_i, x_j) = \kappa_{ij} \max_{\{x' | x'_i = x_i, x'_j = x_j\}} p(\mathbf{x}')$$

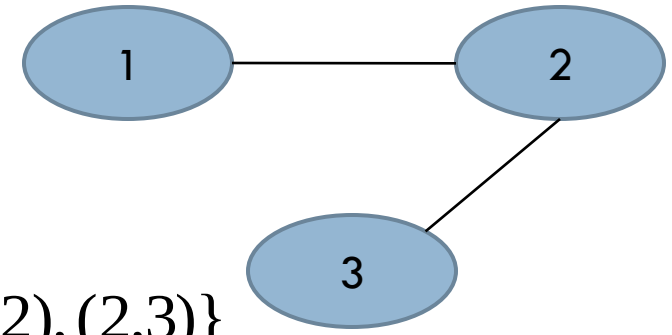
- For any tree distribution $p(\mathbf{x})$ can be factorized into:

$$p(\mathbf{x}) \propto \prod_{i \in v} u_i(x_i) \prod_{(i,j) \in E} \frac{u_{ij}(x_i, x_j)}{u_i(x_i) u_j(x_j)}$$

Max Marginal Example

- Consider 3 residue positions with 2 rotamers each

$$\mathbf{x} \in \{0,1\}^3$$



$$\psi_i(x_i) = 1, \text{ for all } x_i \in \{0,1\} \text{ and } i \in \{1,2,3\}$$

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1 \text{ if } x_i = x_j \\ 4 \text{ otherwise} \end{cases} \quad \text{for all } (i,j) \in \{(1,2),(2,3)\}$$

P({1,1,1}) = 1/50	P({0,0,0}) = 1/50
P({1,1,0}) = 4/50	P({1,0,0}) = 4/50
P({1,0,1}) = 16/50	P({0,0,1}) = 4/50
P({0,1,1}) = 4/50	P({0,1,0}) = 16/50

Max Marginal Example Cont.

P({1,1,1}) = 1/50	P({0,0,0}) = 1/50
P({1,1,0}) = 4/50	P({1,0,0}) = 4/50
P({1,0,1}) = 16/50	P({0,0,1}) = 4/50
P({0,1,1}) = 4/50	P({0,1,0}) = 16/50

$$\max_{\{x' \mid x'_1 = x_1\}} p(\mathbf{x}') = 4^2 / 50 \text{ for } x_1 \in \{0,1\}$$

$$\max_{\{x' \mid x'_i = x_i\}} p(\mathbf{x}') = \frac{4^2}{50} \mu_1(x_1) \text{ so } \mu_1(x_1) = 1 \text{ and } \kappa_1 = 50/4^2$$

The same logic applies to the 2nd and 3rd residue positions

Max Marginal Example Cont.

P({1,1,1}) = 1/50 P({0,0,0}) = 1/50
P({1,1,0}) = 4/50 P({1,0,0}) = 4/50
P({1,0,1}) = 16/50 P({0,0,1}) = 4/50
P({0,1,1}) = 4/50 P({0,1,0}) = 16/50

$$\max_{\{x'|(x'_1,x'_2)=(x_1,x_2)\}} p(\mathbf{x}') = 4/50 \text{ for } x_1 = x_2 \text{ and } 4^2/50 \text{ for } x_1 \neq x_2$$

$$\mu_{ij}(x_i, x_j) = \begin{cases} 1 \text{ if } x_i = x_j \\ 4 \text{ otherwise} \end{cases} \quad \text{for all } (i,j) \in \{(1,2),(2,3)\}$$

$$\max_{\{x'|(x'_1,x'_2)=(x_1,x_2)\}} p(\mathbf{x}') = \frac{4}{50}\mu_{ij}(\mathrm{x_i}, \mathrm{x_j})$$

Using Max Marginal for MAP Assignment

$$p(\mathbf{x}) \propto \prod_{i \in v} u_i(x_i) \prod_{(i,j) \in E} \frac{u_{ij}(x_i, x_j)}{u_i(x_i) u_j(x_j)}$$

$$\max_{\mathbf{x}} p(x) = p(\mathbf{x}^*) = \frac{1}{50} u_1(x_1^*) u_2(x_2^*) u_3(x_3^*) \frac{u_{12}(x_1^*, x_2^*)}{u_1(x_1^*) u_2(x_2^*)} \frac{u_{23}(x_2^*, x_3^*)}{u_2(x_2^*) u_3(x_3^*)}$$

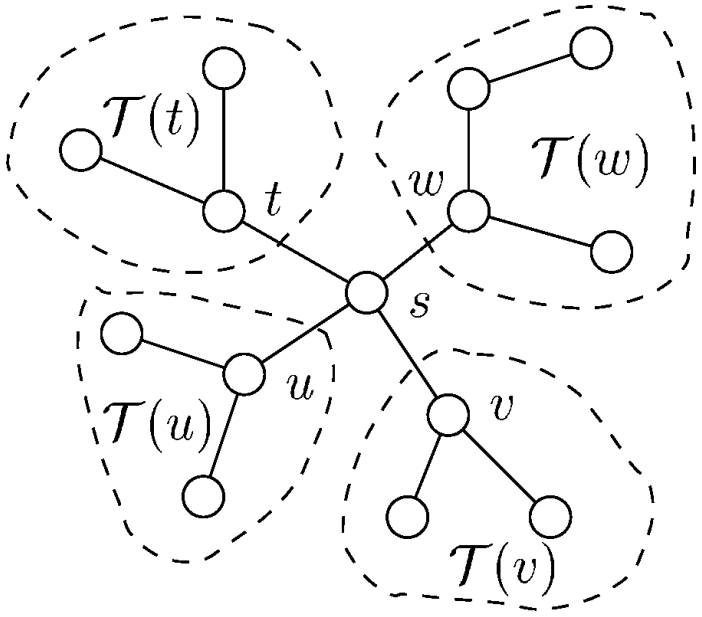
- "Maximum value of p(x) can be obtained simply by finding the maximum value of each $\mu_i(x_i)$ and $\mu_{ij}(x_i, x_i)$"

Max-Product Algorithm

Find max marginal μ_s

$$\mu_s(x_s) = \kappa \psi_s(x_s) \prod_{t \in N(s)} \max_{x'_{T(t)}} \left\{ \psi_{st}(x_s, x'_t)\, p(x'_{T(t)}; \psi_{T(t)}) \right\}$$

$$p(x_{T(t)}; \psi_{T(t)}) \propto \prod_{u \in V(T(t))} \psi_u(x_u) \prod_{(u,v) \in E(T(t))} \psi_{uv}(x_u, x_v)$$

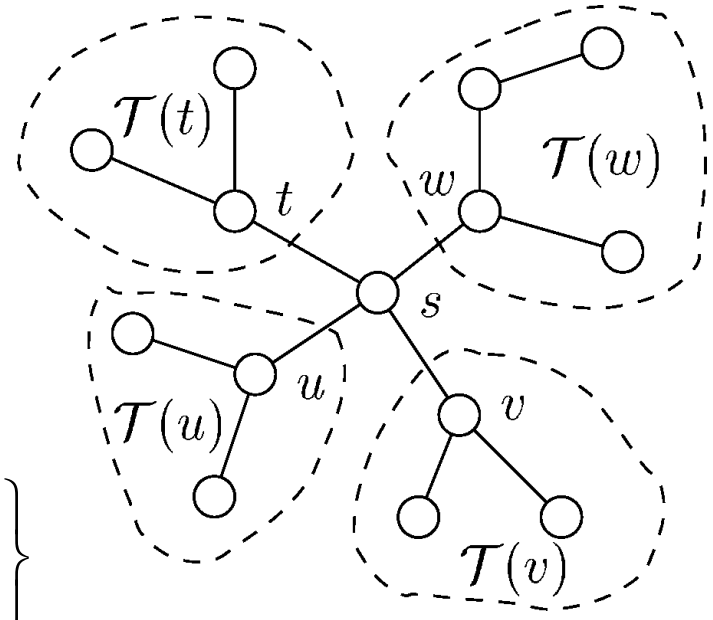


Wainwright, Jaakkola, Willsky. Statistics and Computing. 2004

Max-Product Algorithm

$$\mu_s(x_s) = \kappa \psi_s(x_s) \prod_{t \in N(s)} \max_{x'_{T(t)}} \left\{ \psi_{st}(x_s, x'_t)\, p(x'_{T(t)}; \psi_{T(t)}) \right\}$$

$$p(x_{T(t)}; \psi_{T(t)}) \propto \prod_{u \in V(T(t))} \psi_u(x_u) \prod_{(u,v) \in E(T(t))} \psi_{uv}(x_u, x_v)$$



Node *t* passes message to all of its neighbors *S*

$$M_{ts}^{n+1}(x_s) = \kappa \max_{x'_t} \left\{ \psi_{st}(x_s, x'_t) \psi_t(x'_t) \prod_{u \in N(t)/s} M_{ut}^{n}(x'_t) \right\}$$

$$\tau_s^*(x_s) = \kappa \psi_s(x_s) \prod_{u \in N(s)} M_{us}^*(x_s)$$

$$\tau_{st}^*(x_s, x_t) = \kappa \psi_s(x_s) \psi_t(x_t) \psi_{st}(x_s, x_t) \times \prod_{u \in N(s)/t} M_{us}^*(x_s) \prod_{u \in N(t)/s} M_{ut}^*(x_t)$$

Wainwright, Jaakkola, Willsky. Statistics and Computing. 2004

Max-Product Doesn't work for Cycles

- The algorithm can produce the exact max-marginals for tree-distributions
- Even with exact max-marginals this might not give a MAP solution
- The protein design problem is dense, so there are going to be lots of cycles in the graph
- For general cyclic distributions there is no known method that efficiently computes max-marginals
- We will use pseudo-max-marginals instead

Pseudo-max-marginals

- Break a cyclic distribution into a convex combination of distributions over a set of spanning trees
- Then the pseudo-max-marginals $\nu = \{\nu_i, \nu_{ij}\}$ are defined by construction:

$$p(\mathbf{x}) \propto \prod_{t \in \mathrm{T}} \left[\prod_{i \in \nu} \nu_i(x_i) \prod_{(i,j) \in E} \frac{\nu_{ij}(x_i, x_j)}{\nu_i(x_i)\nu_j(x_j)} \right]^{\rho(t)}$$

- A given tree distribution is

$$p^T(\mathbf{x}; \nu) \propto \prod_{i \in \nu(T)} \nu_i(x_i) \prod_{(i,j) \in E(T)} \frac{\nu_{ij}(x_i, x_j)}{\nu_i(x_i)\nu_j(x_j)}$$

- So total probability is

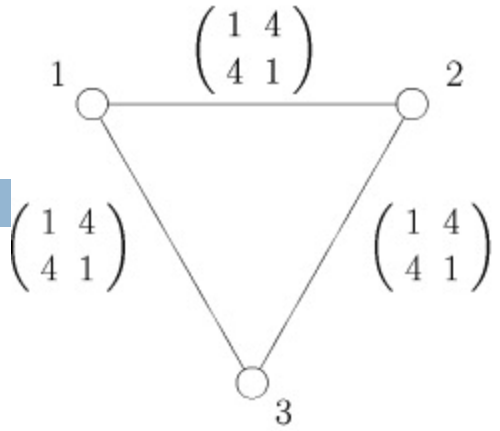
$$p(\mathbf{x}) \propto \prod_{T \in \mathrm{T}} \left[p^T(\mathbf{x}; \nu) \right]^{\rho(t)}$$

Pseudo-Max Marginals

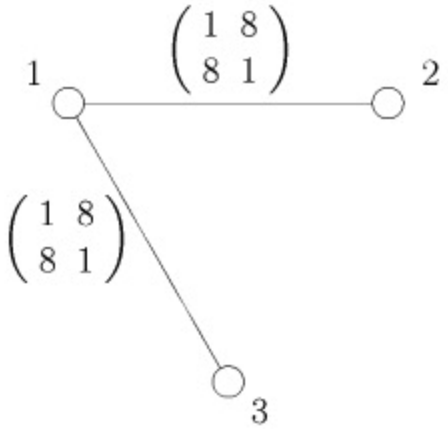
1) ρ-reparameterization

$$p(\mathbf{x}) \propto \prod_{T \in \mathrm{T}} \left[p^T(\mathbf{x}; v) \right]^{\rho(t)}$$

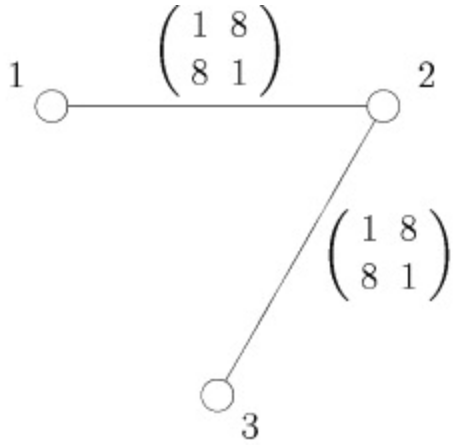
2) Tree consistency



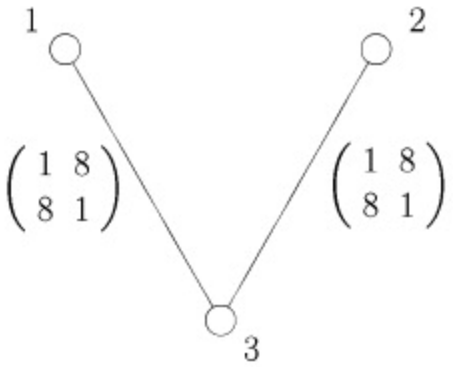
(a) $p(\mathbf{x})$



$p^3(\mathbf{x}; \hat{\nu})$; $\rho^3 = \frac{1}{3}$



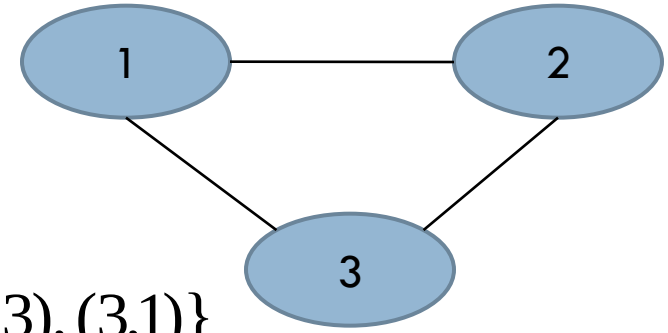
$p^1(\mathbf{x}; \hat{\nu})$; $\rho^1 = \frac{1}{3}$



$p^2(\mathbf{x}; \hat{\nu})$; $\rho^2 = \frac{1}{3}$

Maximal Stars

Pseudo-max marginal example



$$v_i(x_i) = 1, \text{ for all } x_i \in \{0,1\} \text{ and } i \in \{1,2,3\}$$

$$v_{ij}(x_i, x_j) = \begin{cases} 1 \text{ if } x_i = x_j \\ 8 \text{ otherwise} \end{cases} \quad \text{for all } (i,j) \in \{(1,2),(2,3),(3,1)\}$$

$$p^1(\mathbf{x}; v) = v_1(x_1) v_2(x_2) v_3(x_3) \frac{v_{12}(x_1, x_2)}{v_1(x_1) v_2(x_2)} \frac{v_{23}(x_2, x_3)}{v_2(x_2) v_3(x_3)}$$

$$\Psi_i(x_i) = v_i(x_i)^{-1/3} \quad \text{and} \quad \Psi_{ij}(x_i, x_j) = v_{ij}(x_i, x_j)^{2/3}$$

$$p(\mathbf{x}) = \frac{1}{Z} p^1(\mathbf{x}; v)^{1/3} p^2(\mathbf{x}; v)^{1/3} p^3(\mathbf{x}; v)^{1/3}$$

Tree-reweighted max-product algorithm (TRMP)

- Edge-based reparameterization update algorithm to find pseudo max-marginals
- Maintains the ρ-reparameterization criteria
- Upon convergence, satisfies the "tree-consistency condition", that the pseudo-max-marginals converge to the max-marginals of each tree distribution

Bounding GMEC with TRMP

- First require that pseudo-max-marginals obey normal form (i.e. they are all ≤ 1)

$$p(\mathbf{x}) \propto \prod_{T \in \mathrm{T}} \left[p^T(\mathbf{x}; v) \right]^{\rho(t)}$$

$$\max_x p(\mathbf{x}) = \max_x \frac{v_c}{Z} \prod_{S \in \mathbf{S}} \left[p^S(\mathbf{x}; v) \right]^{\rho(S)} \leq \frac{v_c}{Z} \prod_{S \in \mathbf{S}} \left[\max_x p^S(\mathbf{x}; v) \right]^{\rho(S)}$$

$$\max_x p(\mathbf{x}) \leq \frac{v_c}{Z} \Rightarrow \min_x \mathrm{e}(\mathbf{x}) \geq -\ln(v_c)$$

$$p(\mathbf{x}) = \frac{1}{Z} \exp[-e(\mathbf{x})]$$

Bounding conformation with given rotamer

$$\max_{\{x|x_\zeta = r\}} p(\mathbf{x}) = \max_{\{x|x_\zeta = r\}} \frac{v_c}{Z} \prod_{S \in \mathbf{S}} \{p^S(\mathbf{x}; v)\}^{\rho(S)}$$

$$\leq \frac{v_c}{Z} \prod_{S \in \mathbf{S}; \zeta \in V(S)} \{\max_{\{x|x_\zeta = r\}} p^S(\mathbf{x}; v)\}^{\rho(S)} \times \prod_{S \in \mathbf{S}; \zeta \notin V(S)} \{\max_{\{x|x_\zeta = r\}} p^S(\mathbf{x}; v)\}^{\rho(S)}$$

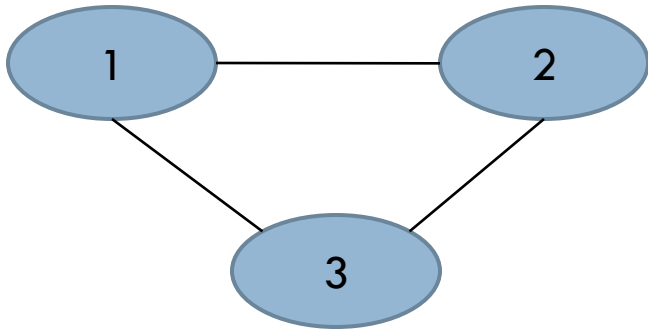
$$\kappa_\zeta \max_{x_\zeta \in R_\zeta} p^S(\mathbf{x}; v) = v_\zeta(r) \qquad \max_{x_\zeta \in R_\zeta} p^S(\mathbf{x}; v) = \max_{\mathbf{x}} p^S(\mathbf{x}; v) = 1$$

$$\max_{\{x|x_\zeta = r\}} p(\mathbf{x}) \leq \frac{v_c}{Z} v_\zeta(r)^{\rho_\zeta}$$

Back to Pseudo-Max-Marginal Example

$$p(\mathbf{x}) \propto \prod_{T \in \mathrm{T}} \left[p^T(\mathbf{x}; v) \right]^{\rho(t)}$$

P({1,1,1}) = 1/98 P({0,0,0}) = 1/98
P({1,1,0}) = 16/98 P({1,0,0}) = 16/98
P({1,0,1}) = 16/98 P({0,0,1}) = 16/98
P({0,1,1}) = 16/98 P({0,1,0}) = 16/98



- Bound Energy: $p^s([0,0,0]) = \frac{1}{8}\left(\frac{1}{8}\right)$ for $s \in \{1,2,3\}$

$$\max_x p(\mathbf{x}) \leq \frac{v_c}{Z} \qquad \frac{1}{98} = \frac{v_c}{98}\left[\left(\frac{1}{8}\right)\frac{1}{8}\right]^{3^{1/3}} \Rightarrow v_c = 64$$

- Bound Rotamer:

$$\max_{\{x | x_\zeta = r\}} p(\mathbf{x}) \leq \frac{v_c}{Z} v_\zeta(r)^{\rho_\zeta} \qquad \frac{v_c}{Z} v_1(0)^{1/3} = \frac{64}{98}$$

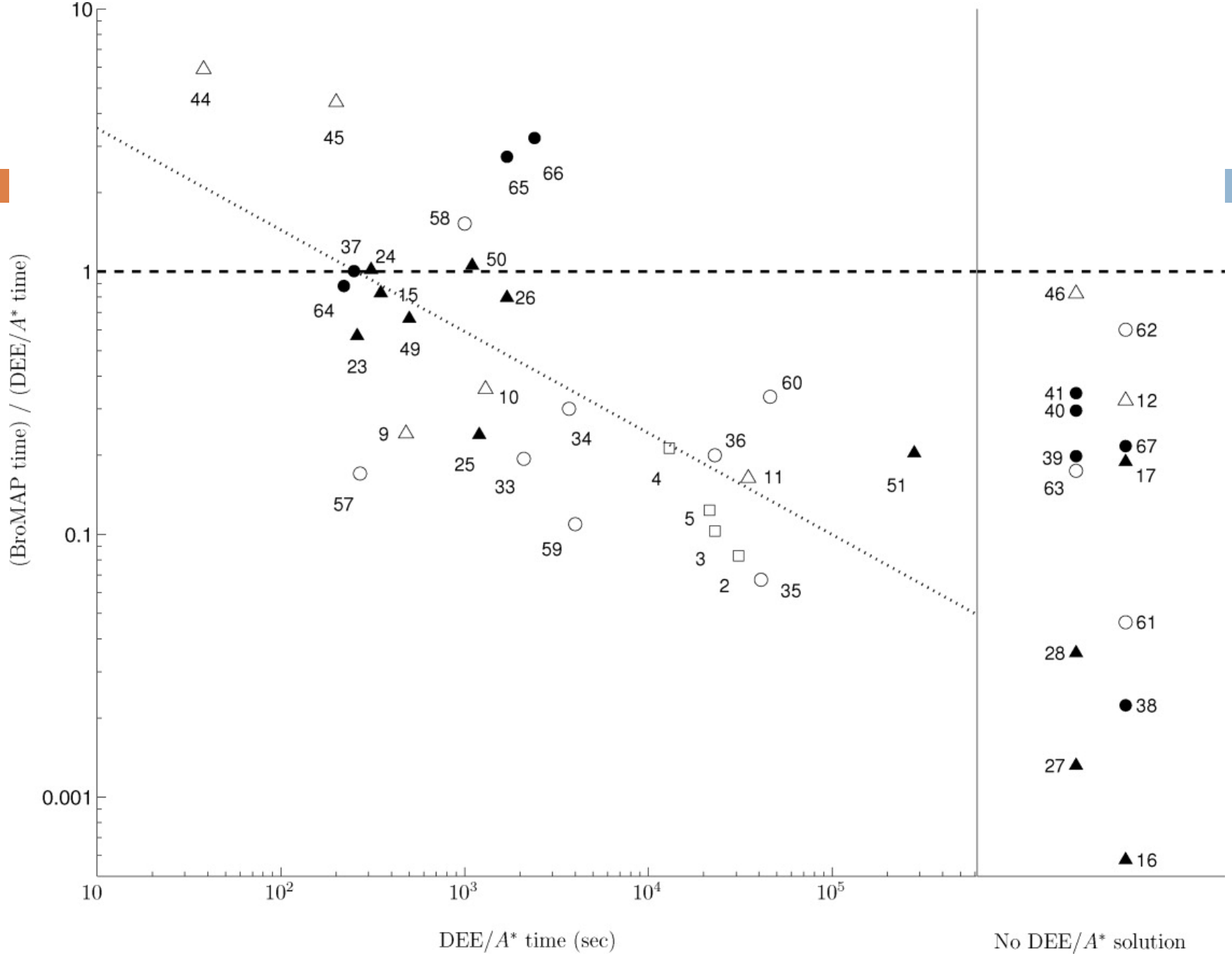
Summary of Bounding Subproblems

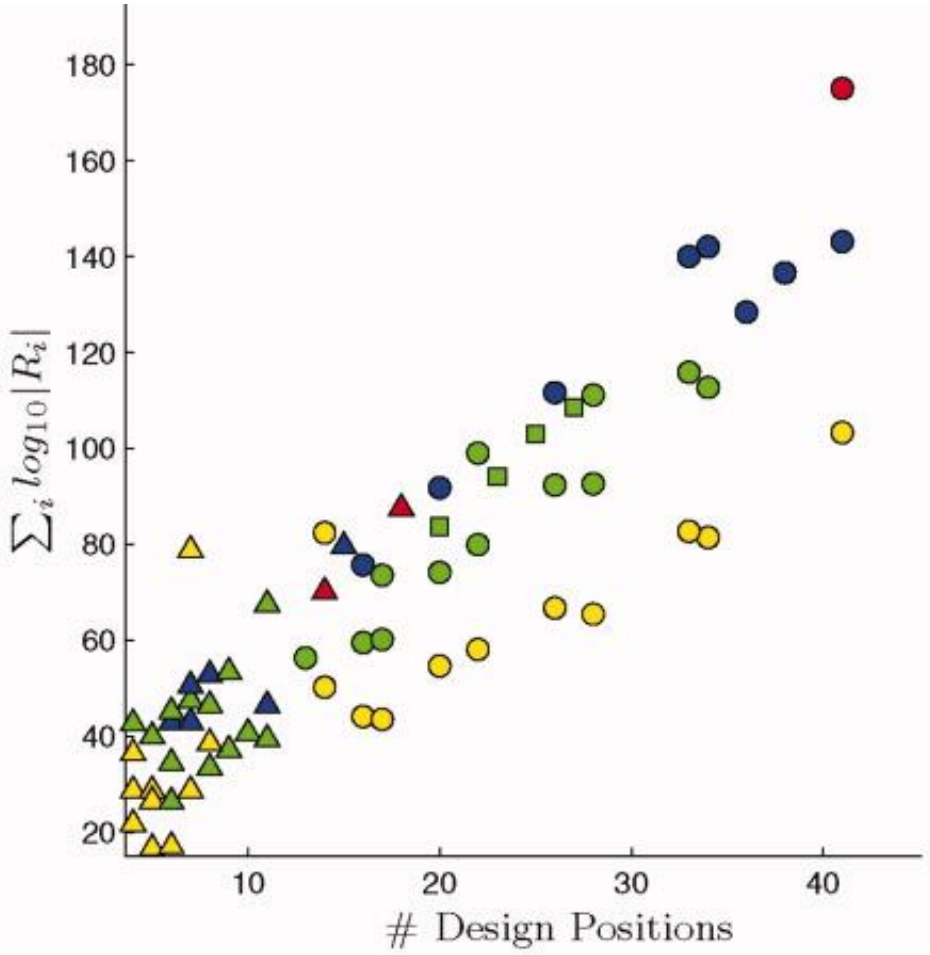
- Started with max-marginals, but they didn't work for cycles so moved to pseudo-max-marginals
- Break up full cycle graph into stars, and then use TRMP to find pseudo-max-marginals
- Since pseudo-max-marginals are in normal form and tree consistent, we can use them to bound the actual

Results

- Test cases:
  - FN3: 94-residue B-sheet
  - D44.1 and D1.3: Antibodies that bind hen egg-white lysozyme
  - EPO: Human erythropoeitin complexed with receptor
- Ran DEE/A* and BroMAP (their algorithm) and allowed 7 days to finish
- DEE/A* solved 51 cases, BroMAP solved 65 out of 68 total cases

No.	Bro	DEE	T-Br	F-Br	Skew	F-Ub	Leaf	Rdctn	RC	%DE	%A*	%TR
2	2.6 E 3	3.1 E 4	31	25	0.90	0.49	30.7	2.12	36	42.8	0.3	56.3
3	2.4 E 3	2.3 E 4	31	26	0.93	0.49	27.7	2.55	32	46.2	0.6	52.6
4	2.8 E 3	1.3 E 4	23	23	1	0	33.7	3.01	0	43.9	0.3	55.5
5	2.7 E 3	2.1 E 4	26	26	1	0.55	27.4	3.12	0	37.2	0.4	62.2
9	1.2 E 2	4.8 E 2	3	3	1	0	27.6	1.93	0	8.9	74.1	17.0
10	4.6 E 2	1.3 E 3	13	10	0.75	0.37	26.9	1.02	74	7.6	70.4	14.4
11	5.7 E 3	3.5 E 4	109	17	0.81	0.36	26.2	0.85	663	3.8	78.9	11.2
15	2.9 E 2	3.5 E 2	0	0	NA	0	NA	NA	0	94.6	0.4	4.7
23	1.5 E 2	2.6 E 2	0	0	NA	0	NA	NA	0	86.7	0	12.6
24	3.2 E 2	3.1 E 2	4	4	1	0	25.3	4.33	0	62.3	15.1	21.6
25	2.9 E 2	1.2 E 3	0	0	NA	0	NA	NA	0	89.6	0	10.4
26	1.4 E 3	1.7 E 3	11	11	1	0.89	29.2	1.65	0	46.1	0.4	53.2
33	4.1 E 2	2.1 E 3	13	13	1	0	27.9	2.43	0	34.7	4.5	59.8
34	1.1 E 3	3.7 E 3	19	19	1	0	30.0	2.32	0	32.2	2.7	64.8
35	2.8 E 3	4.1 E 4	21	21	1	0	28.7	3.03	0	50.7	0.6	48.6
36	4.6 E 3	2.3 E 4	25	25	1	0	27.9	3.39	0	53.2	0.7	45.9
37	2.5 E 2	2.5 E 2	0	0	NA	0	NA	NA	0	76.0	2.4	21.2
44	2.2 E 2	3.8 E 1	8	6	0.71	0.54	28.2	1.87	17	8.2	75.5	14.1
45	8.8 E 2	2.0 E 2	8	8	1	0	26.2	5.16	8	48.6	23.8	25.4


10
1
0.1
0.001
(BroMAP time) / (DEE/A* time)
10
10^2
10^3
10^4
10^5
DEE/A* time (sec)
No DEE/A* solution
44
45
66
65
58
37
24
50
64
15
26
49
23
10
60
9
34
36
25
33
4
11
51
57
5
59
3
2
35
46
62
41
12
40
67
39
17
63
61
28
38
27
16



(a) Solvability

INT

CORE

CORE++

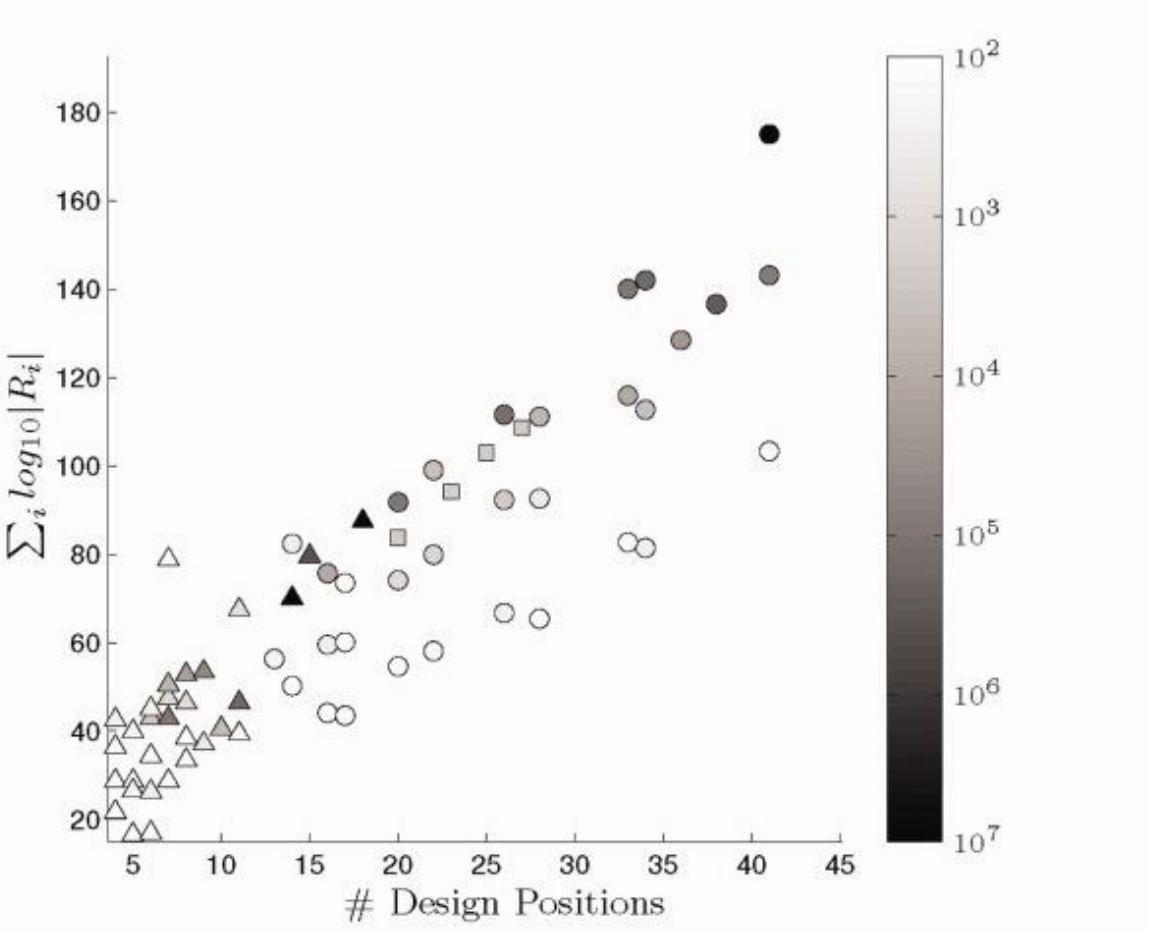
Solved By

Limited DEE

BroMAP and DEE

BroMAP only

Nothing



Protein region and BroMAP running time

Conclusions

- Exact solution approach for large, dense protein design problems
- Solved harder problems faster than DEE/A* and solved some that DEE/A* couldn't
- Performance advantage:
  - Smaller search trees
  - Can perform additional elimination and informed branching from inexpensive lower bounds

Comparison to DACS

- Both partition on residue position in order to increase pruning and reduce A* search tree
- BroMAP makes an *informed* partition
- BroMAP uses intermediate information to prune partial conformations
- BroMAP loses ability to enumerate “in order”