AND/OR Branch-and-Bound for Computational Protein Design Optimizing K*

Bobak Pezeshki (pezeshkb@uci.edu), Radu Marinescu (radu.marinescu@ie.ibm.com),
Alex Ihler (ihler@ics.uci.edu), Rina Dechter (dechter@ics.uci.edu)

PAPER VIDEO

Overview

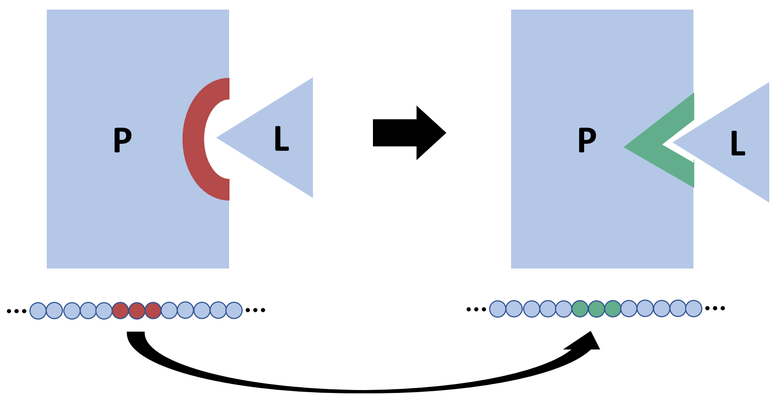
The objective of this work is to **(1)** create a framework for future advancements for protein design via K* optimization (ie. K*MAP) leveraging powerful MMAP algorithms, **(2)** explore new related bounded heuristics, and **(3)** create a foundation for efficient algorithms solving such protein design problems.

Contributions:

1. Formulation of K*MAP as a graphical model
2. wMBE-K*, a weighted mini-bucket scheme for K*MAP enhanced with a domain partitioning scheme
3. AOBB-K*, a depth-first branch-and-bound algorithm over AND/OR search spaces for solving K*MAP
4. A thresholding scheme introducing and exploiting determinism with correctness guarantees
5. Extensive analysis comparing these schemes to state-of-the-art BBK* illustrating their potential

Problem

Redesign of proteins to form higher affinity complexes



Find new amino acid assignment to residues of interest that optimize affinity between interacting subunits

K* Objective

An approximation of binding affinity between molecules (based on the biological association constant known as K_a)

$$K^*(r) = \frac{Z_{PL}(r)}{Z_P(r)\, Z_L(r)}$$

$$Z_\gamma(r) = \sum_{c \in C_\gamma(r)} \exp\{-E_\gamma(c)/\mathcal{R}\mathcal{T}\}$$

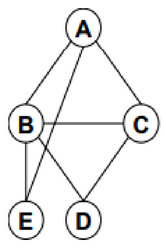
captures the goodness of the subunit(s) in form $\gamma \in \{P, L, PL\}$

r = amino acid assignments to residues
C(r) = possible conformations given r
E(c) = energy given conformation c
$\mathcal{R}$ = universal gas constant
$\mathcal{T}$ = absolute temperature (Kelvin)

AND/OR Search

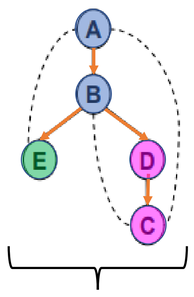
Compact search space taking advantage of conditional independences present in the model

Graphical Model Network



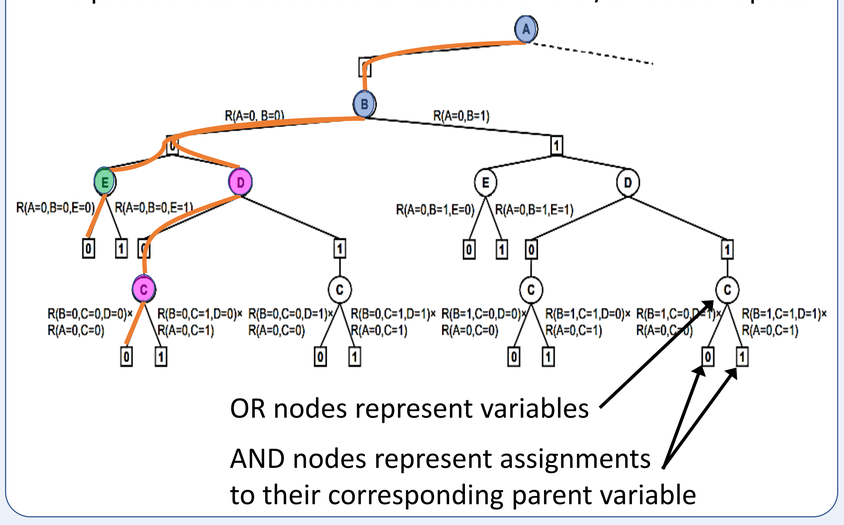
R(AB)
R(AC)
R(ABE)
R(BCD)

Possible Pseudo Tree



Directed tree (based on a variable ordering) that branches when conditional independences exist given assignments to ancestors.

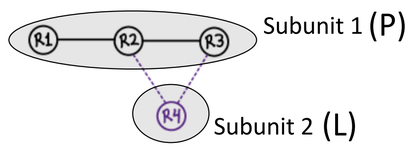
The pseudo tree is used to construct the AND/OR search space



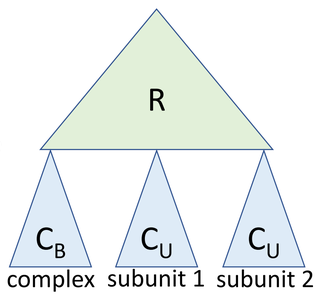
OR nodes represent variables

AND nodes represent assignments to their corresponding parent variable

Graphical Model Formulation

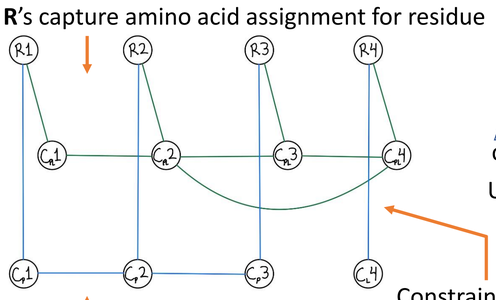


$$\mathcal{M} = < \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F} >$$



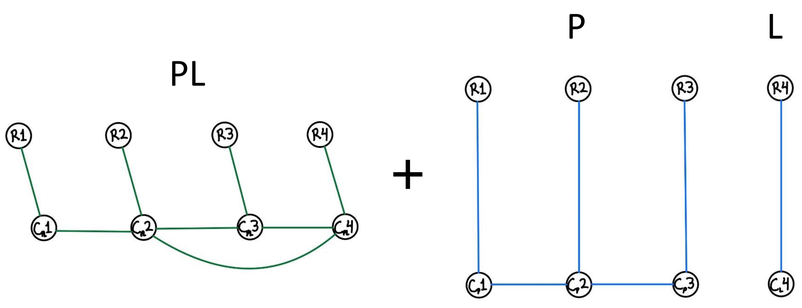
Unbound Form: U = {P} ∪ {L}
Bound Form: B = {PL}

R's capture amino acid assignment for residue



Constraints between corresponding **C**'s and **R**'s ensure consistent assignments

C's capture all (amino acid, rotamer) combinations possible at its corresponding R



K*MAP Task

Find amino acid assignments to the residues that maximize K*

$$Z_\gamma(R_1 \ldots R_N) = \sum_{C_1,\ldots,C_N} \prod_{\mathscr{C}_{\gamma(i)} \in \mathscr{C}} \mathscr{C}_{\gamma(i)}(R_i, C_{\gamma(i)}) \cdot \prod_{E^{sb}_{\gamma(i)} \in \boldsymbol{E}^{sb}_\gamma} e^{-\frac{E^{sb}_{\gamma(i)}(C_{\gamma(i)})}{\mathscr{R}T}} \cdot \prod_{E^{pw}_{\gamma(ij)} \in \boldsymbol{E}^{pw}_\gamma} e^{-\frac{E^{pw}_{\gamma(ij)}(C_{\gamma(i)}, C_{\gamma(j)})}{\mathscr{R}T}}$$

$$K^*(R_1, \ldots, R_N) = Z_B(R_1, \ldots, R_N) / Z_U(R_1, \ldots, R_N)$$

$$task: \; K^*MAP = \max_{R_1,\ldots,R_N} K^*(R_1, \ldots, R_N)$$

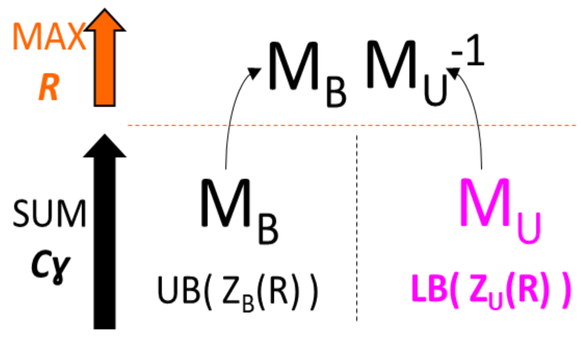
wMBE-K*

Precompiled K* heuristic to guide search, based on dynamic programming message-passing Mini Bucket Elimination

$$K^*(r) = \frac{Z_{complex}(r)}{Z_{subunit\ 1}(r)\, Z_{subunit\ 2}(r)}$$

$$\sum_x^w [\prod_r (\psi_r)] \le \prod_r^{w_r} [\sum_x]$$

$$\sum_x^w f(x) \triangleq [\sum_x f(x)^{\frac{1}{w}}]^w \qquad w = \sum_r w_r$$



Domain-Partitioned wMBE-K*

Strategy to improve lower bounds generated by wMBE-K*

- Let X, Y, and Z be three variables
- Let $obj = \sum_X f(x,y) \cdot g(x,z)$
- Let $X' = \{x \in X | g(x,z) \neq 0\}$ s.t. $\epsilon_{X'} = \min_{x \in X'} g(x,z)$
- Since $\epsilon_{X'} > 0$, we can derive…

$$obj = \sum_X f(x,y) \cdot g(x,z)$$

$$obj = \sum_{x \in X'} f(x,y) \cdot g(x,z) + \sum_{x \in X \setminus X'} f(x,y) \cdot g(x,z)$$

$$obj = \sum_{x \in X'} f(x,y) \cdot g(x,z) \ge \epsilon_{X'} \cdot \sum_{x \in X'} f(x,y) > 0$$

AOBB-K*

- ❑ Branch-and-bound algorithm over AND/OR search spaces
  - AOBB-K* is exact
- ❑ Can use wMBE-K* to guide search
- ❑ Exploits determinism by using constraint propagation
- ❑ Incorporates a global constraint enforcing biologically relevant solutions

Subunit Stability Constraints

Condition to enforce the stability of each subunit to be no less than a given threshold from that of the wild-type stability

$$Z_{subunit\ i}(r) > Z_{subunit\ i}(r^{wt}) * \exp\{-5/\mathcal{R}\boldsymbol{T}\}$$

Stability of naturally occurring version — Constant factor for thresholding

Infusing Determinism: τ-Underflows

Replace extremely unfavorable assignments with hard constraints taking exploiting the strength of constraint propagation

- Let f be a non-negative function
- Consider $\tau \in \mathbb{R}^+$
- Then the τ underflow of f is…

$$f(x) = \begin{cases} f(x), & f(x) \ge \tau \\ 0, & otherwise \end{cases}$$

Empirical Analysis

Competing Scheme: BBK*

[Ojewole et al., 2018]

State-of-the-art protein redesign algorithm, part of the software package, OSPREY, developed for over ten years for protein design

• A*-like best first • utilizes dynamic optimistic greedy heuristic
• approximate scheme w/ tightness parameter

Empirical Results

Dataset (#instances)	AOBB-K* K*≥ (F1,F2)	AOBB-K* K*> (F1,F2)	AOBB-K* time< (F1,F2)	any-AOBB-K* K*≥ (F1,F2)	any-AOBB-K* K*> (F1,F2)
Orig. (30)	30,30	2,2	23,28	30,30	2,2
Expand. (12)	6,11	0,4	2,4	11,11	1,4

benchmarks (3 MAP Variables)	ω	τ	(203 ≤ Dmax ≤ 206) iB	w*	d	|X|	UB	pre-t	[ω,τ]-AOBB-K* search	time	K*	BBK* time	K*
	-	-	4	4	7	13	28.8	123.8	81.3	205.1			
1gwc_00021*	0.001	-	4	4	7	13	28.8	124.3	12.1	136.4	11.9	551.3	11.7
	-	1E-05	4	4	7	13	28.8	117.1	3.5	120.7			
	-	-	4	6	9	17	42.3	109.8	44.0	153.8			
2hnv_00025*	0.001	-	4	6	9	17	42.3	109.3	12.2	121.5	16.2	880.5	13.6
	-	1E-05	4	6	9	17	42.3	100.8	1.7	102.4			
	-	-	4	6	9	17	37.7	83.0	17.8	100.8			
2rf9_00013*	0.001	-	4	6	9	17	37.7	82.8	1.6	84.4	15.0	39.2	15.0
	-	1E-05	4	6	9	17	37.7	71.4	0.4	71.8			
	-	-	4	5	8	15	34.1	58.3	2.6	60.9			
2rfe_00012*	0.001	-	4	5	8	15	34.1	58.6	0.3	58.9	13.9	11.8	13.9
	-	1E-05	3	5	8	15		5.5	14.9	20.3			
	-	-	4	5	8	15	35.1	58.2	2.4	60.6			
2rfe_00014*	0.001	-	4	5	8	15	35.1	58.2	1.3	59.4	14.4	44.9	14.4
	-	1E-05	3	5	8	15		4.9	15.2	20.0			
	-	-	5	5	8	15	26.4	166.8	167.8	334.6			
2rfe_00017*	0.001	-	4	5	8	15	27.4	89.1	5.0	94.1	10.9	78.0	10.8
	-	1E-05	4	5	8	15	27.4	85.7	7.1	92.8			
	-	-	4	5	8	15	31.3	115.2	161.5	276.6	11.1		
2rfe_00030*	0.001	-	4	5	8	15	31.3	101.4	0.5	101.9	10.9	275.4	11.0
	-	1E-05	4	5	8	15	30.4	105.4	3.5	108.9	11.1		
	-	-	5	5	8	15	26.2	81.8	278.9	360.7			
2xgy_00020*	0.001	-	4	5	8	15	27.2	60.4	8.2	68.6	11.0	1388.1	11.0
	-	1E-05	4	5	8	15	27.2	60.1	23.8	83.9			
	-	-	4	4	7	13	11.4	62.6	36.8	99.5			
3u7y_00009*	0.001	-	4	4	7	13	11.4	62.5	2.1	64.7	4.5	215.8	4.5
	-	1E-05	4	4	7	13		61.3	5.1	66.4			
	-	-	4	4	7	13	28.3	74.0	2.1	76.1	11.9		
3u7y_00011*	0.001	-	4	4	7	13	28.3	83.4	0.1	83.5		26.6	11.9
	-	1E-05	4	4	7	13	28.3	80.7	0.5	81.2			
	-	-	5	5	8	15	37.0	169.2	12.1	181.3	15.0		
4wwi_00019*	0.001	-	4	5	8	15	38.0	62.0	7.2	69.2		34.0	15.0
	-	1E-05	4	5	8	15		54.2	8.4	62.7			

Summary

- ❑ AOBB-K* shows promise vs. state-of-the-art BBK*
  - ❑ Competitive run-times
  - ❑ Can find better solutions
- ❑ τ-AOBB-K* can greatly improve runtimes
- ❑ AOBB-K* has scalability issues as the number of MAP variables increase

Acknowledgements

Special Thanks:
- Thomas Schiex [INRAE, France]
- Bruce Donald and members of his lab [Duke University] including: Graham Holt, Jonathan Jou and Nathan Guerin

Supported in part by: NSF Grant IIS-2008516