Protein Pow(d)er

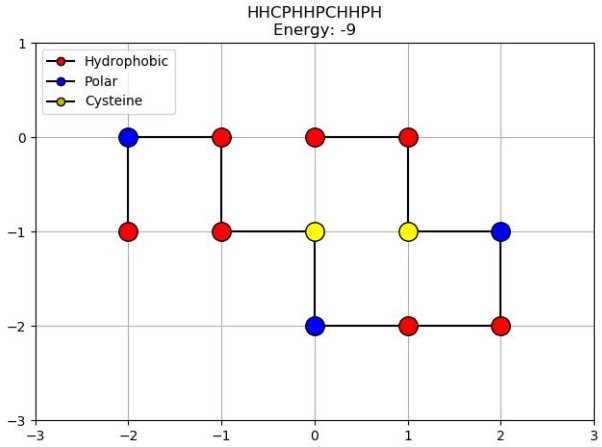
The legend of Cysteine

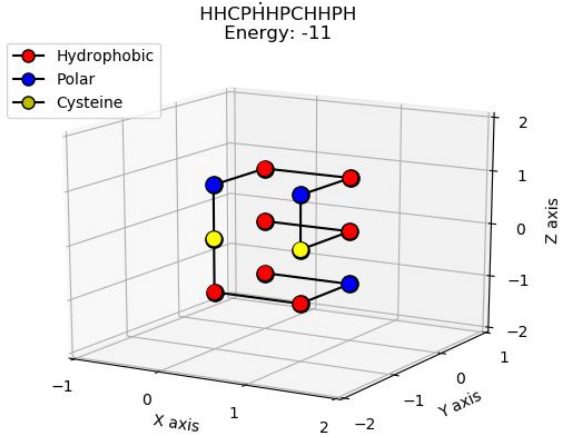
Ruby Bron
Michael Stroet
Sophie Stiekema

De case

- Eiwitten vouwen
- Aminozuren op een 2D / 3D rooster
  - **Hydrofoob**
  - **Polair**
  - **Cysteïne**

- Minimaliseren van energie
  - **P** → 0
  - **H - H** → - 1
  - **H - C** → - 1
  - **C - C** → - 5

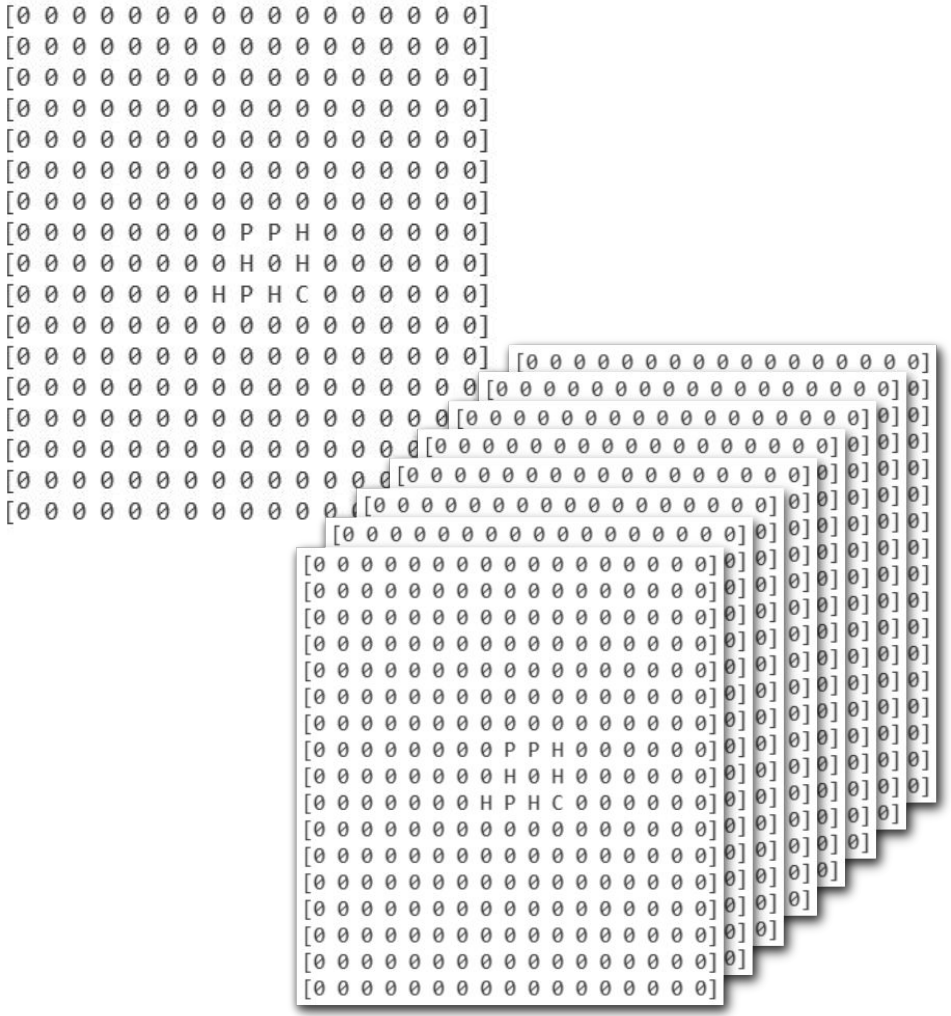




Eiwitten vouwen

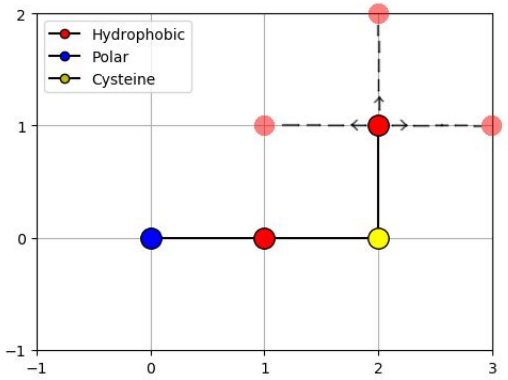
- 2D: Matrix (lijst van lijsten)
- 3D: lijst van matrices

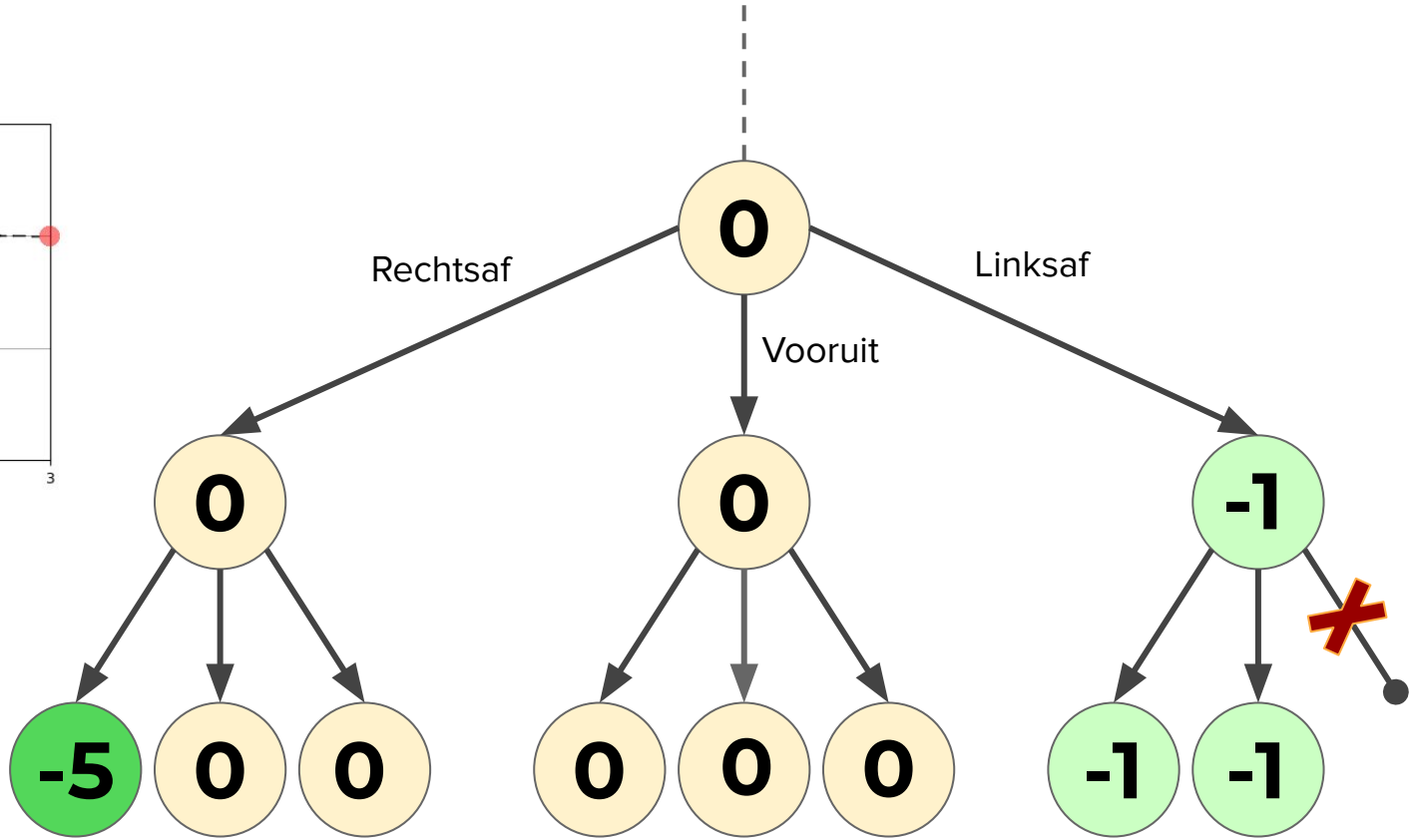
- Matrix grootte varieerbaar



Toestandsruimte

➔ P H C H H C

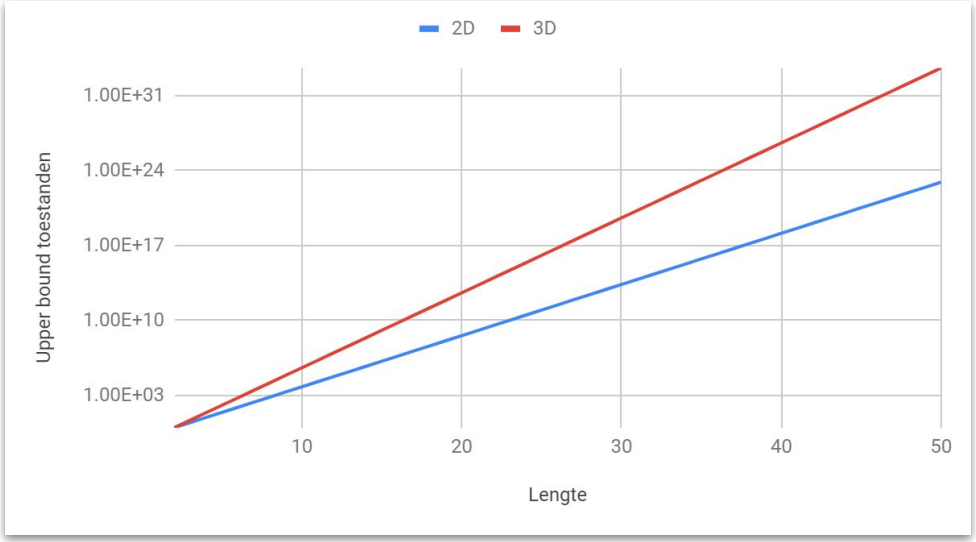




Toestandsruimte

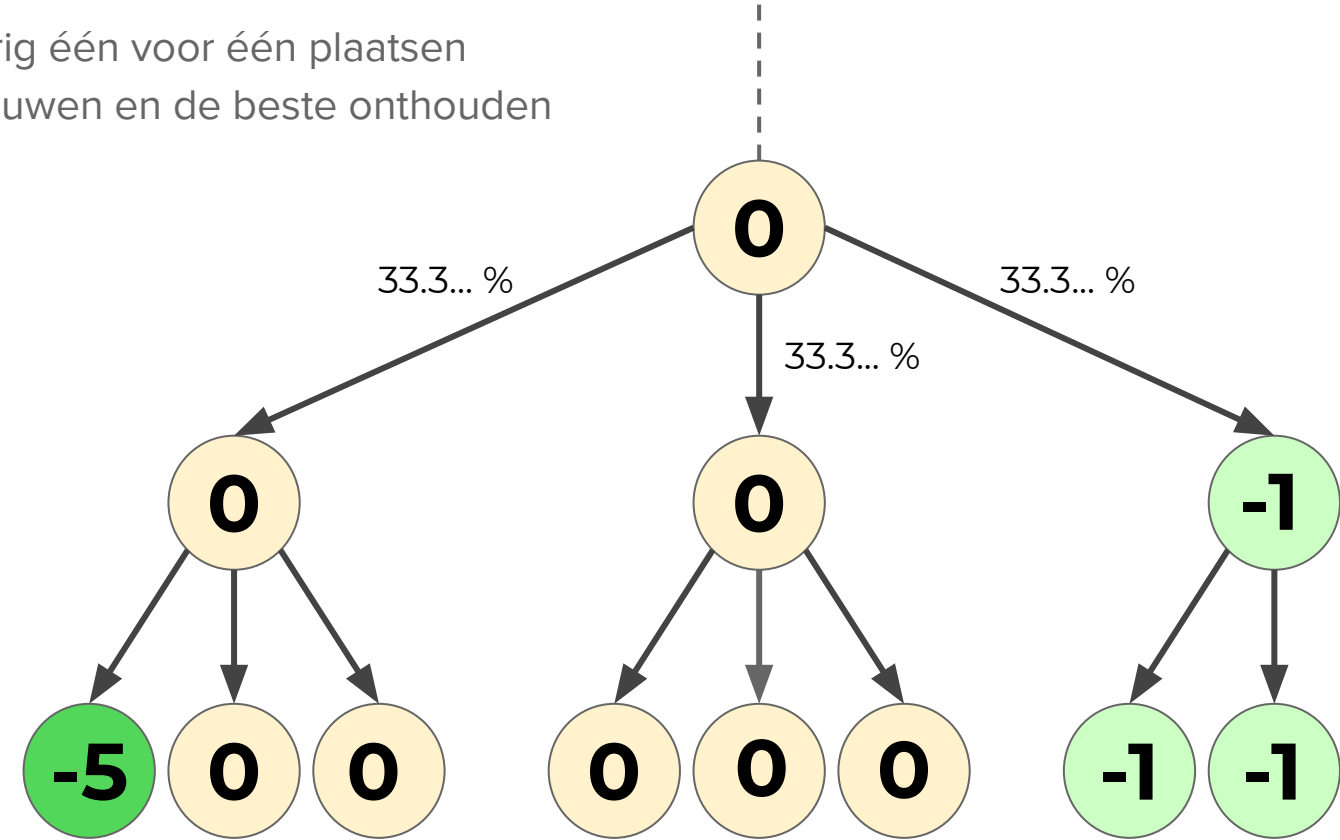
- 2D: $\mathbf{3}^{\text{lengte - 2}}$ lengte 50: $3^{48} \approx 8.0 * 10^{22}$
- 3D: $\mathbf{5}^{\text{lengte - 2}}$ lengte 50: $5^{48} \approx 3.6 * 10^{33}$

- Toestandsruimte verkleinen:
  - Symmetrie
  - Matrix grootte



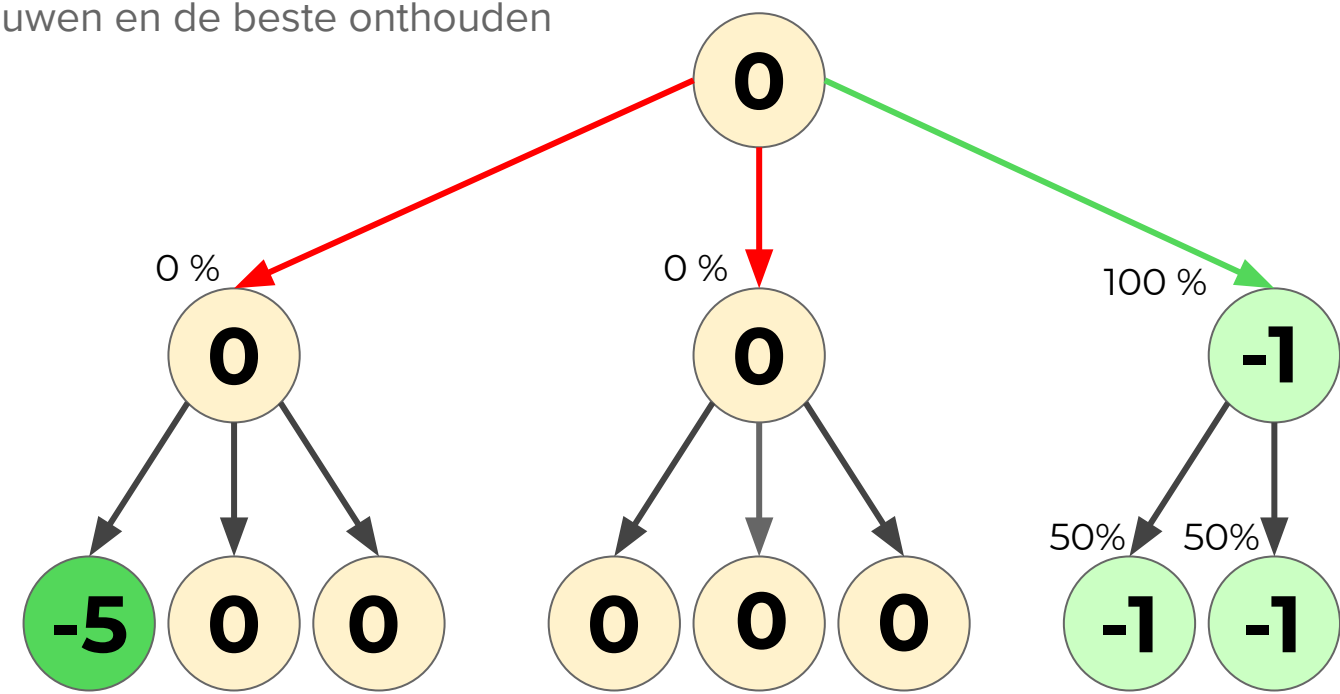
Random Walk

- Aminozuren willekeurig één voor één plaatsen
- N aantal proteïnen vouwen en de beste onthouden



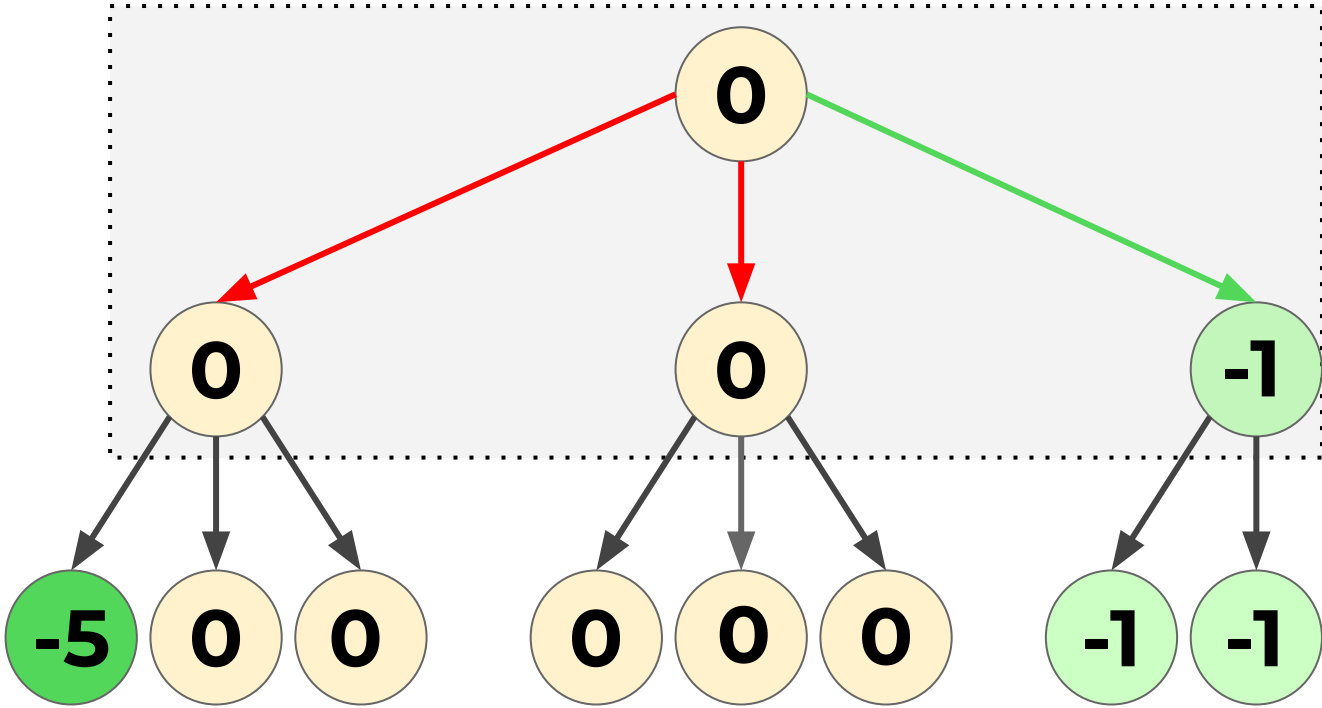
Greedy (look-ahead)

- Aminozuren één voor één plaatsen bij de beste energie
  - Bij gelijke energie waarde, willekeurige keuze tussen deze richtingen
- N aantal proteïnen vouwen en de beste onthouden



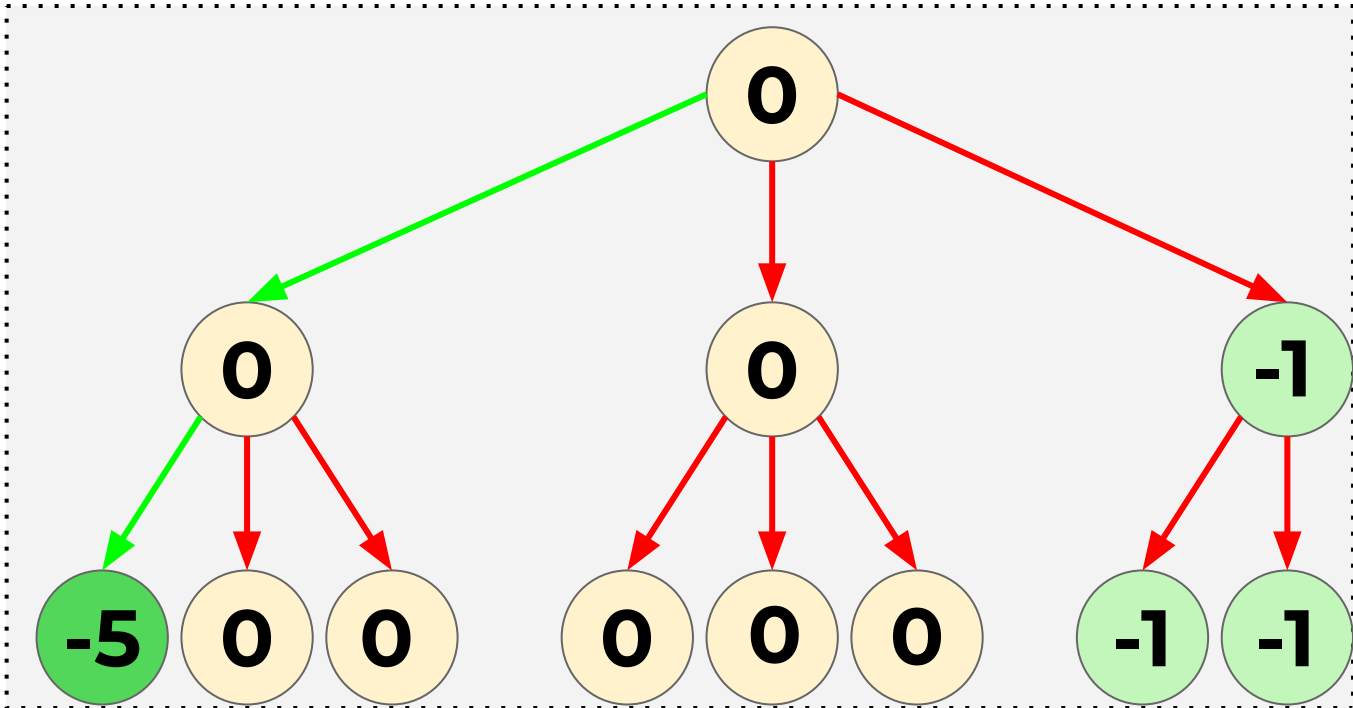
Greedy (look-ahead)

➔ Zonder look-ahead



Greedy (look-ahead)

➔ Met look-ahead



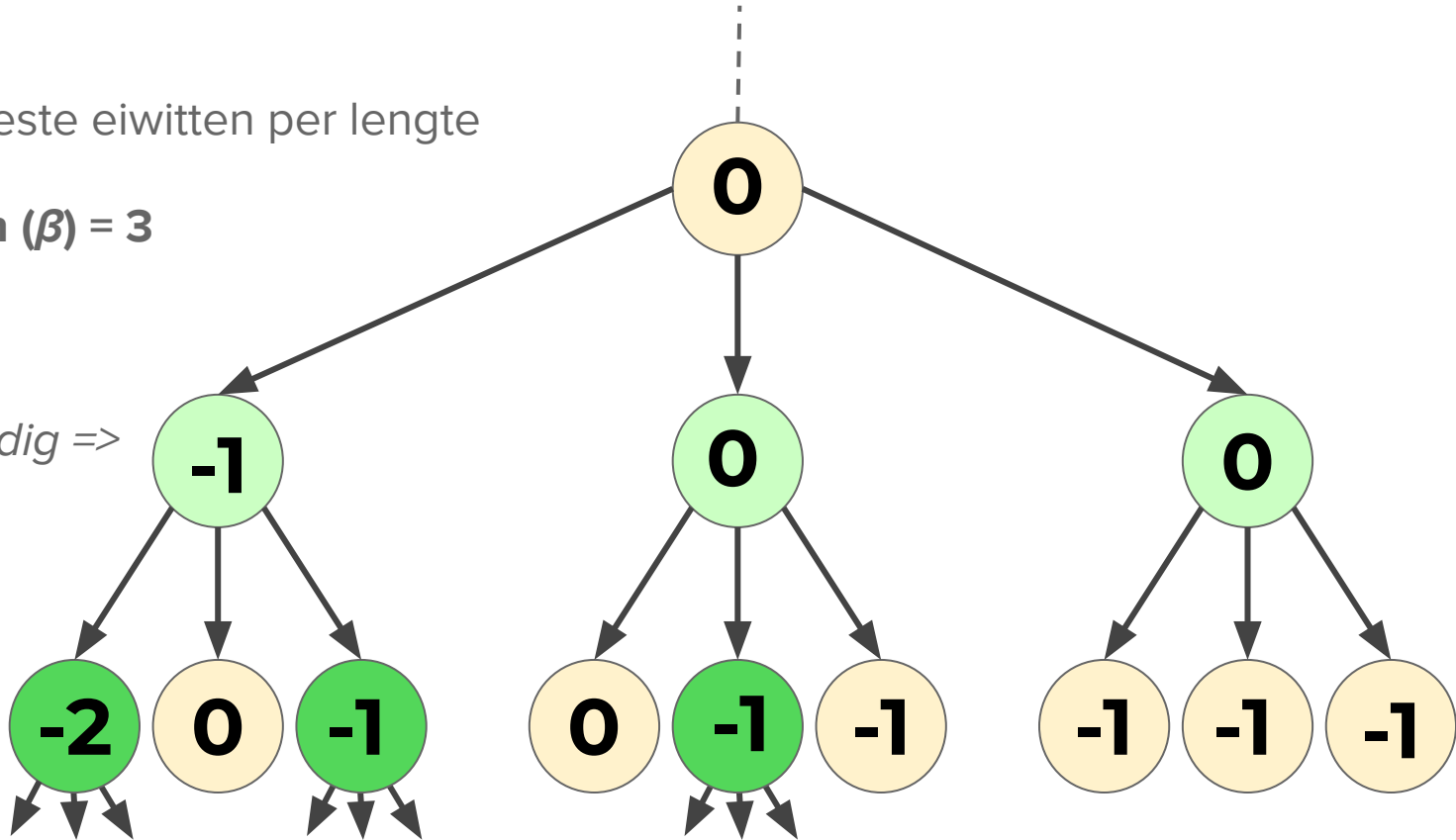
Constructief: Beam Search

- Breadth-first
- Onthoudt β beste eiwitten per lengte

➝ **Beam width (β) = 3**

Beam width oneindig =>

Exact algoritme



Constructief: Branch ‘n Bound

- *Depth-first*, recursief, non-stack en probability based
- Aminozuren een voor een plaatsen en energie bijhouden

<u>Hogere energie dan gemiddelde?</u>

- Kleine kans om verder te gaan

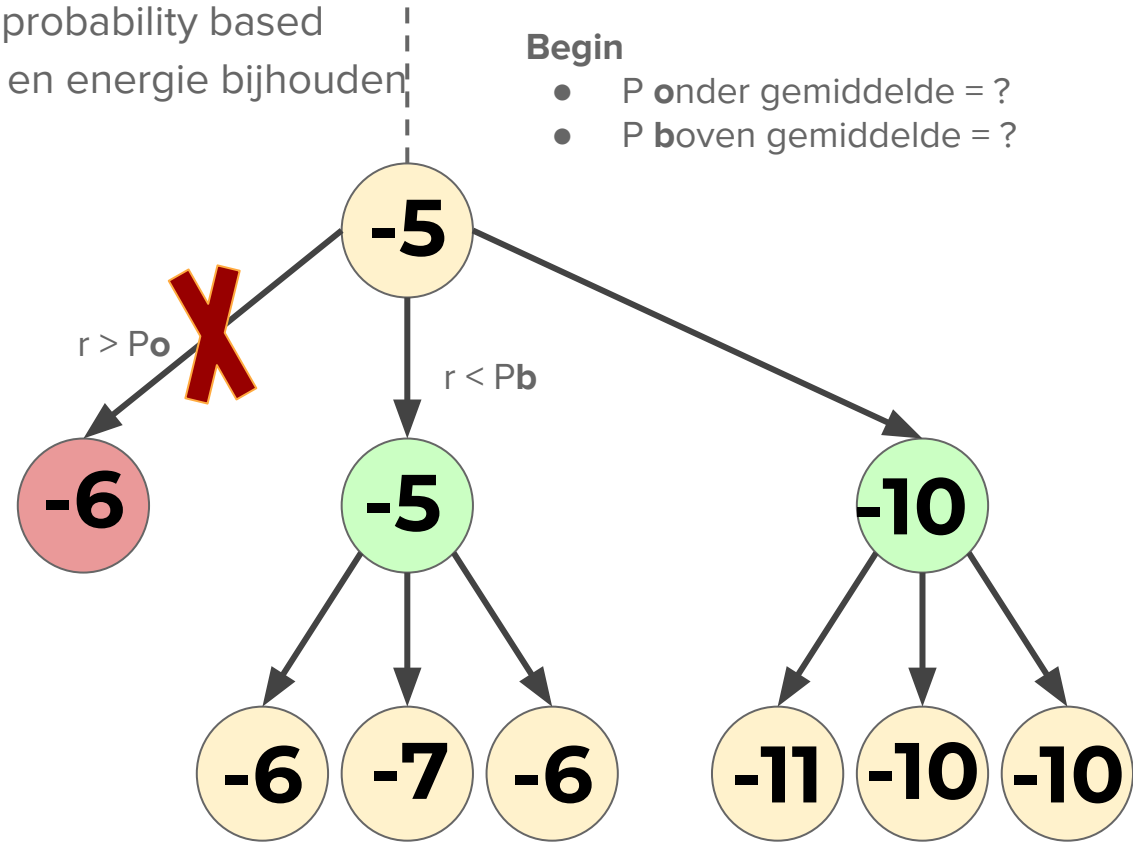
<u>Lagere energie dan gemiddelde?</u>

- Kans om verder te gaan

<u>Laagste energie tot nu toe?</u>

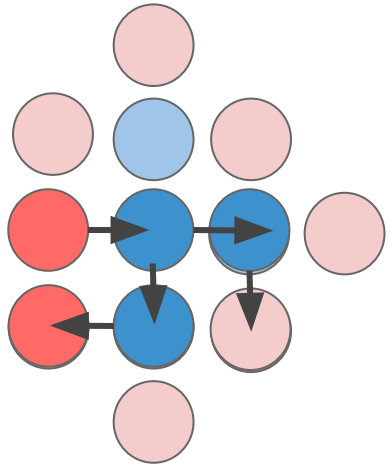
- Verder gaan

Kansen 1 & 1 => Exact algoritme



Iteratief: Hill Climber (Local Search)

- Start eiwit:
  - Greedy
- n - iteraties
  - Wegknippen
  - Terugplaatsen
  - Energie testen
    - Hoger: weigeren
    - Lager/gelijk: accepteren



energie: 0

Beste resultaten

	Proteïne	Lengte	2D	3D
1	HHPHHHPH	**8**	-3	-3
2	HHPHHHPHPHHHPH	**14**	-6	-7
3	HPHPPHHPHPPHPHHPPHPH	**20**	-9	-11
4	PPPHHPPHHPPPPPHHHHHHHHPPHHPPPPHHPPHPP	**36**	-14	-18
5	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH	**50**	-21	-30

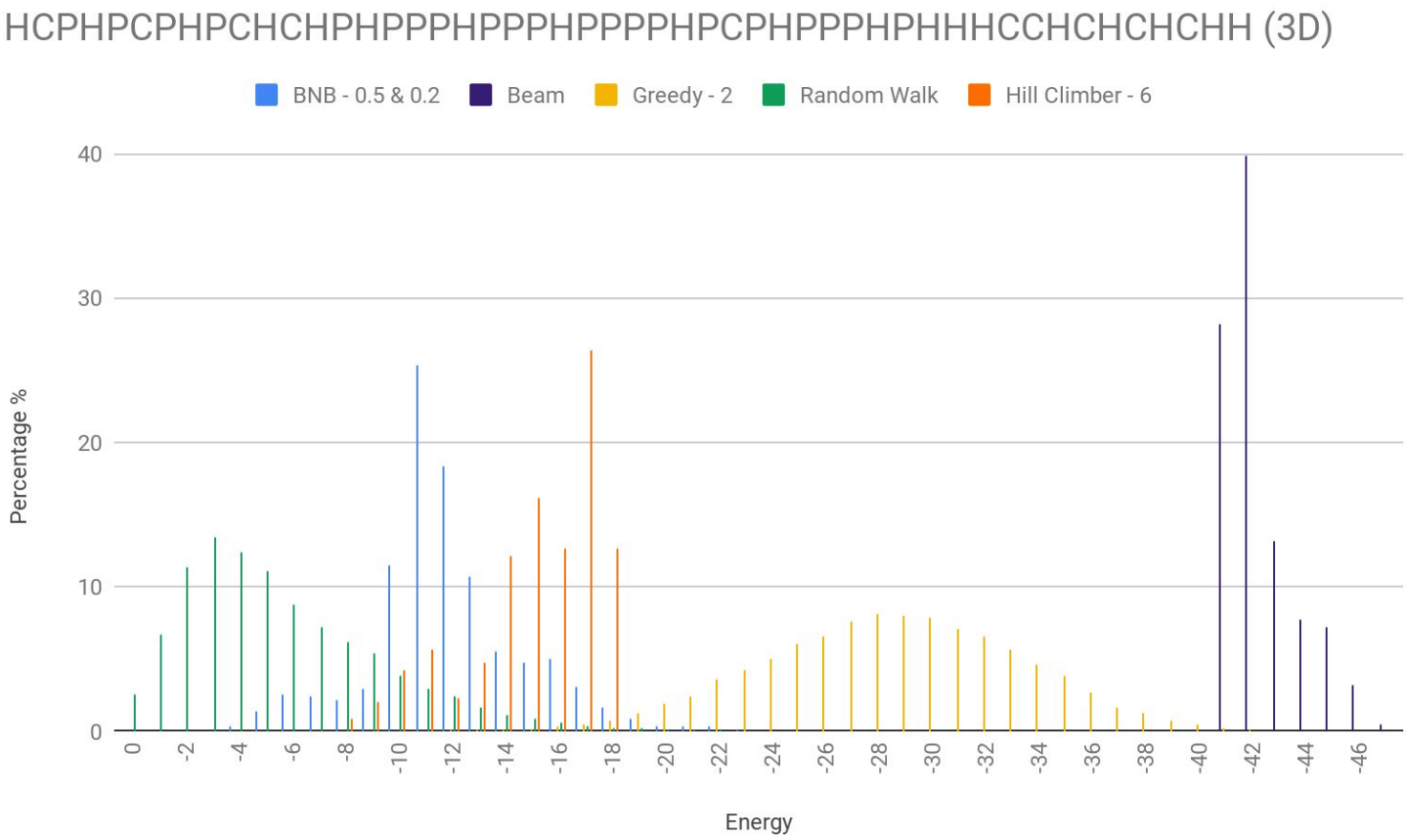
Energiën per algoritme (2D, lengte/2, ~10 minuten)

	Lengte	Random walk 1.000.000	Greedy 12.500 | 3	Beam Search 10.000	Branch 'n bound ~ 10 min	Hill Climber 4x 5000 | 5
1	**8**	-3	-3	-3	-3	-3
2	**14**	-6	-6	-6	-6	-6
3	**20**	-8	-8	-9	-9	-9
4	**36**	-12	-13	-13	-13	-8
5	**50**	-14	-20	-21	-20	-13
6	**36**	-21	-25	-23	-21	-18
7	**36**	-32	-37	-38	-36	-27
8	**50**	-24	-28	-29	-28	-24
9	**50**	-27	-33	-34	-33	-23

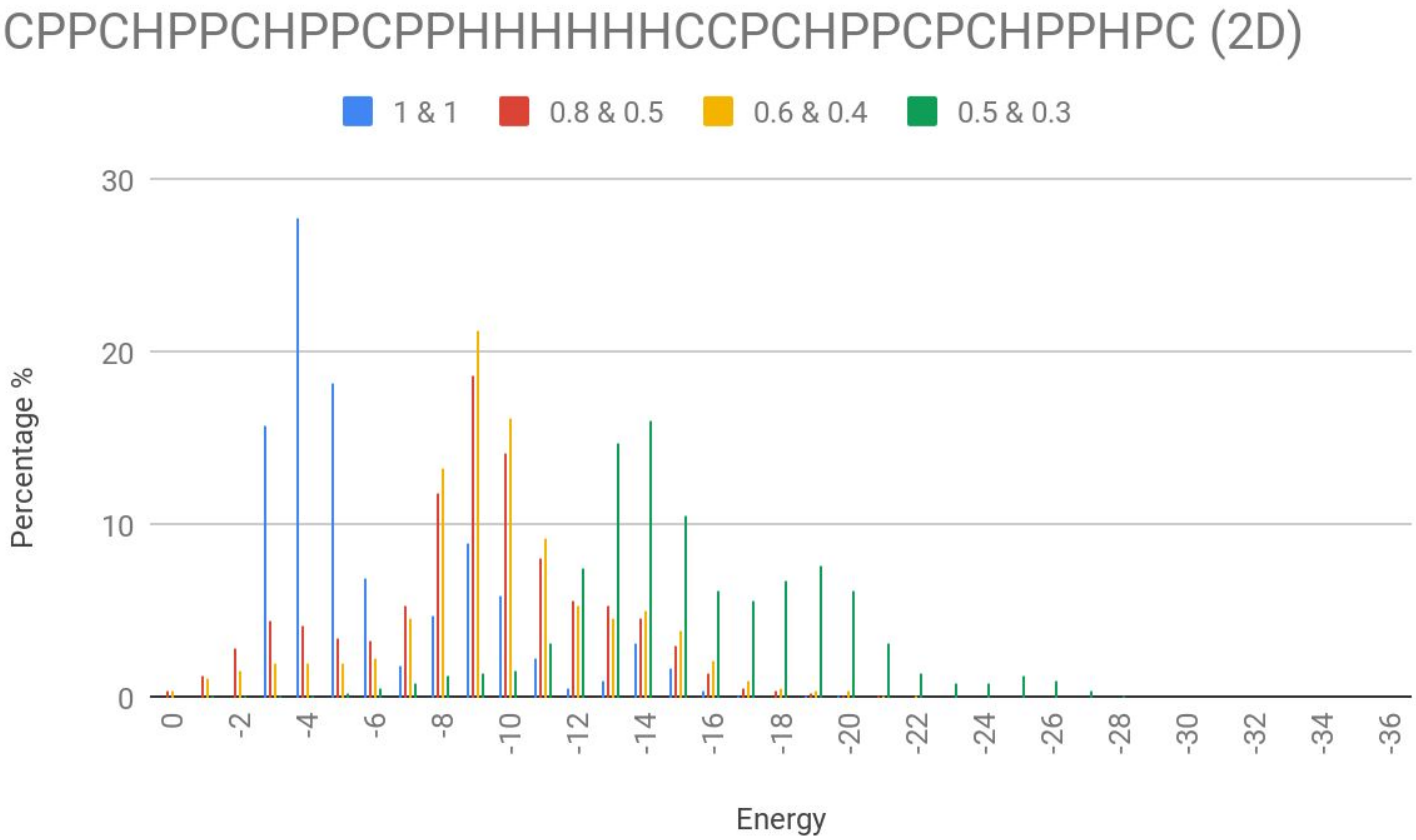
Energiën per algoritme (3D, lengte/3, ~20 minuten)

	Lengte	Random walk 1.000.000	Greedy 15.000 | 2	Beam Search 6000	Branch 'n bound ~ 20 min	Hill Climber 4x 3000 | 5
1	**8**	-3	-3	-3	-3	-3
2	**14**	-7	-7	-7	-7	-7
3	**20**	-9	-11	-11	-10	-11
4	**36**	-14	-18	-18	-18	-12
5	**50**	-18	-30	-30	-21	-21
6	**36**	-26	-35	-35	-33	-21
7	**36**	-38	-58	-57	-52	-39
8	**50**	-35	-47	-47	-35	-31
9	**50**	-27	-50	-53	-48	-39

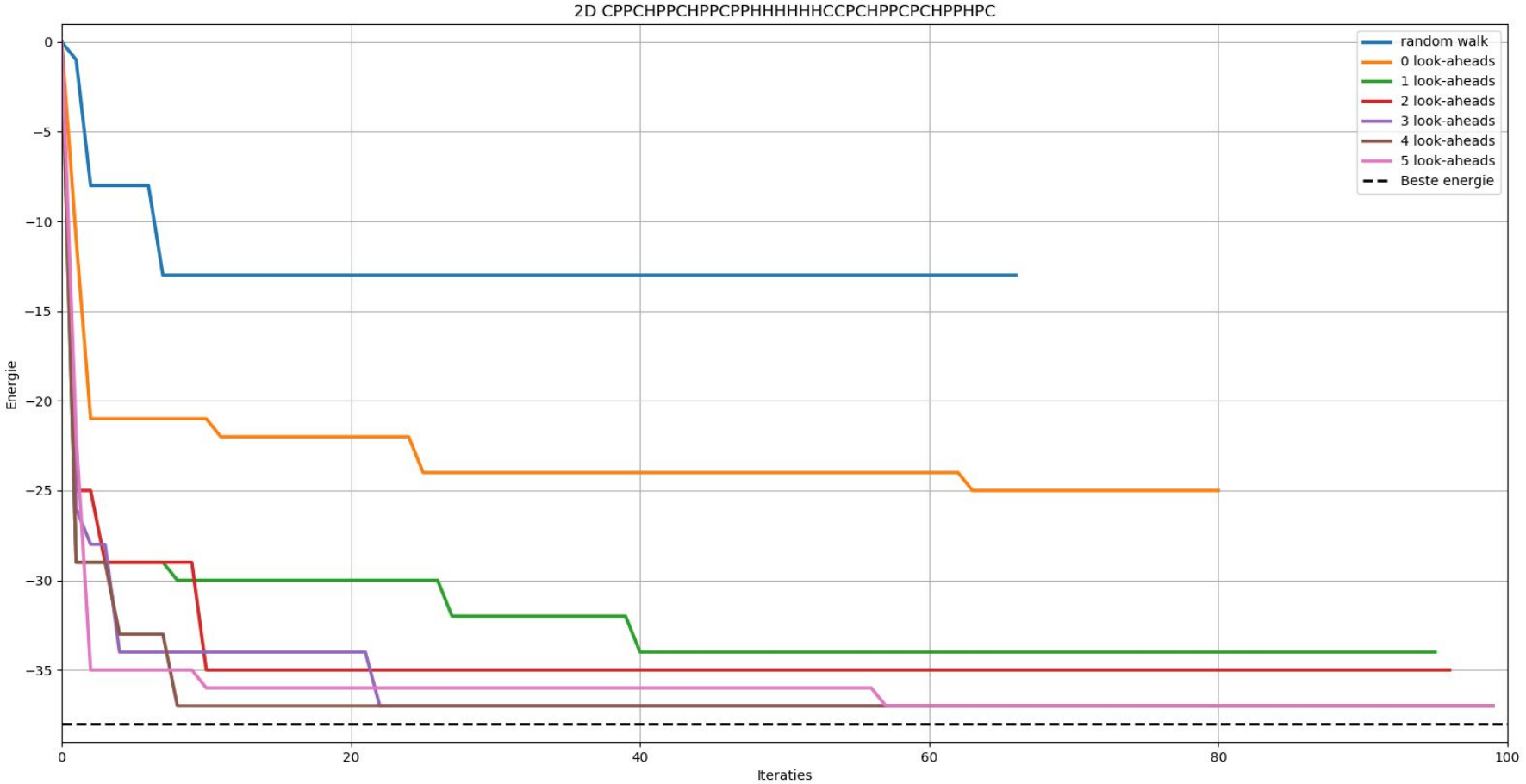
Vergelijking - Cysteine 3D



BnB - probabilities vergelijking - 15min

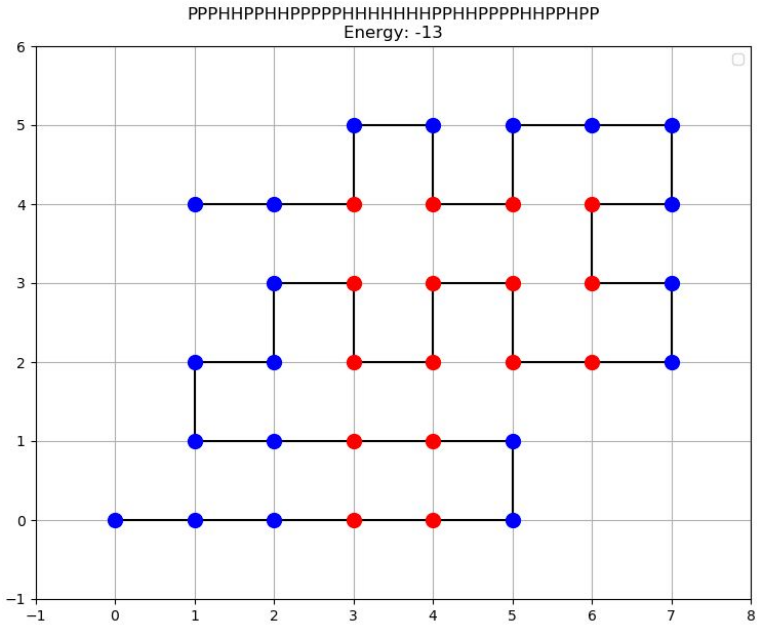


Vergelijking - Random walk en Greedy

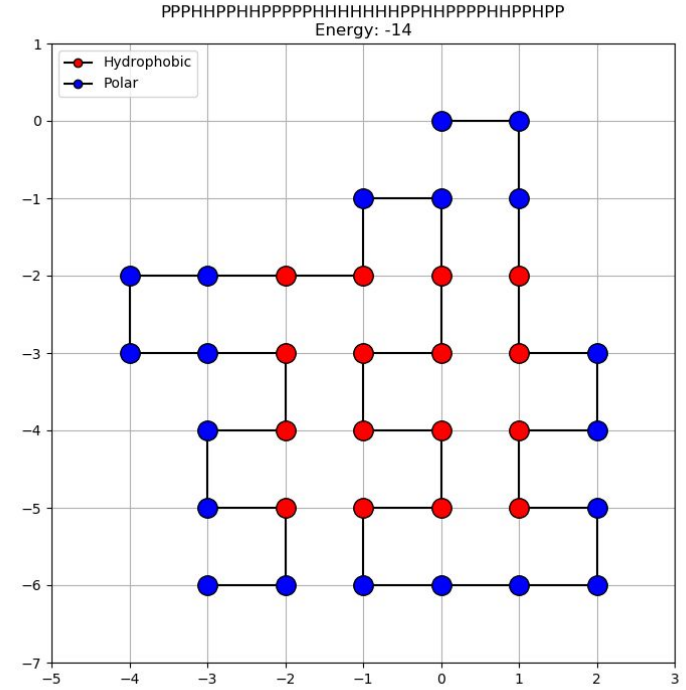


Vouwing proteïne 4 (2D)

Branch ’n bound, 0.75 & 0.25,
00:14:22,
Matrix grootte: 2 * lengte - 1

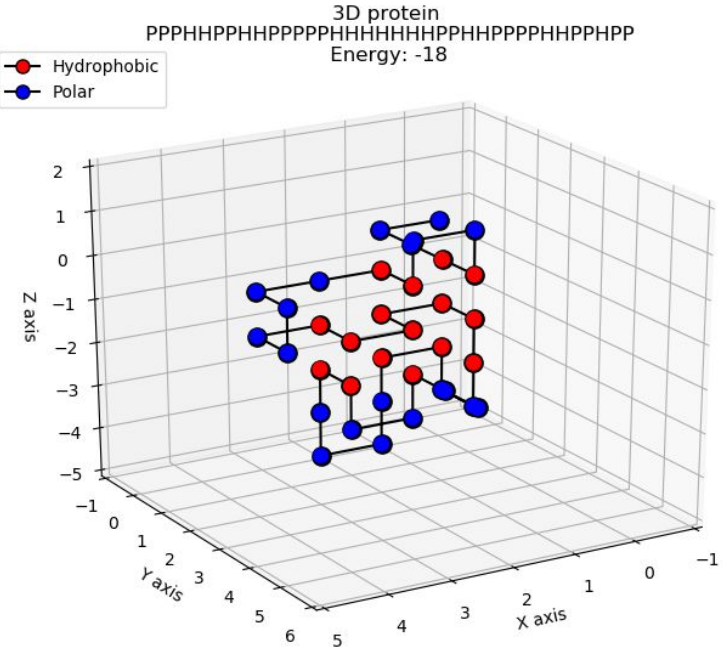


Look-ahead 2, 10.000 proteïnes,
00:02:04,
Matrix grootte: lengte / 2



Vouwing proteïne 4 (3D)

Look-ahead 2, 1.500 proteïnes,
00:01:19,
Matrix grootte: lengte / 3



Discussie

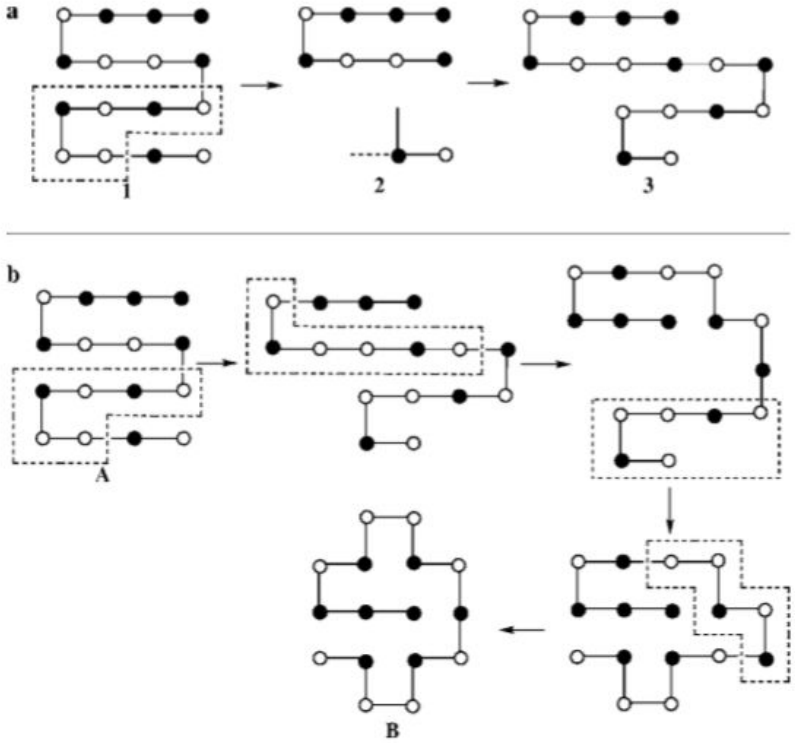
- **Greedy**: Meer look-aheads is niet altijd beter
- **Branch ‘n Bound**: Rekentijd neemt heel snel toe
- **Hillclimber**: Zit snel vast in een local minimum
    - Niet altijd veranderingen
    - Begin situatie bepalend voor eindscore

Conclusie

- **Random Walk:** snel, maar geeft slechte oplossingen
- **Greedy met Lookahead:** snel een goede oplossing
- **Beam Search:** langzamer**,** meer geheugen, gelijke of betere oplossing
- **Branch ‘n Bound:** traag, mogelijkheid tot veel betere oplossingen
- **Hill Climber:** niet optimaal, oplossing sterk afhankelijk van begin eiwit

Toekomst suggesties

- Beam Search met Lookahead
- Hillclimber optimalisatie
- Simulated annealing
    - Zang, Kou & Liu (2007)



Referenties

Chen, M. & Huang, W. (2005). A Branch and Bound Algorithm for the Protein Folding Problem in the HP Lattice Model. *Genomics, Proteomics & Bioinformatics,* 3(4), 225-230.

Zhang, Jinfeng, Samuel C. Kou, and Jun S. Liu. (2007). *Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo*. Journal of Chemical Physics 126(22): 225101.

Vergelijking - Random walk en Greedy