Experimentatie

**Statespace:**

2D: 3^(n-2)

3D: 5^(n-2)

Dit zijn de *theoretish* maximaal behaalbare verschillende vouwingen.

Deze statespace is als volgend bepaald:

De eerste twee aminozuren kun je gewoon standaard vast leggen, omdat het niet uitmaakt welke richting je opgaat hiermee. Dit geeft je een standaard mogelijkheid

* 1

Daarna kun je in 2D, 3 verschillende kanten op vouwen en in 3D 5 verschillende kanten.

* 1 + 3

Dit kan daarna ook met welke aaneenvolgende aminozuren

* 1 + 3^(n)

Op de eerste twee aminozuren na, want die staan al vast...

* 1 + 3 ^(n-2)

|  |  |  |
| --- | --- | --- |
| **Keten** | **Lower-bound** | **Upper-bound (2D)** |
| **HHPHHHPHPHHHPH** | **1** | **531,442** |
| **HPHPPHHPHPPHPHHPPHPH** | **1** | **387,420,490** |
| **PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP** | **1** | **16,677,181,699,666,570** |
| **HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH** | **1** | **239,299,329,230,617,529,084** |
| **CPPCHPPCHPPCPPHHHHHHCCPCHPPCPCHPPHPC** | **1** | **16,677,181,699,666,570** |
| **HCPHPCPHPCHCHPHPPPHPPPHPPPPHPCPHPPPHPHHHCCHCHCHCHH** | **1** | **239,299,329,230,617,529,084** |
| **HCPHPHPHCHHHHPCCPPHPPPHPPPPCPPPHPPPHPHHHHCHPHPHPHH** | **1** | **239,299,329,230,617,529,084** |

**Score**

Berekening Upper-bound:

* (aantal H – 1) \* 1 + (aantal C - 1) \* 5

|  |  |  |
| --- | --- | --- |
| **Keten** | **Lower-bound** | **Upper-bound** |
| **HHPHHHPHPHHHPH** | **0** | **-9** |
| **HPHPPHHPHPPHPHHPPHPH** | **0** | **-9** |
| **PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP** | **0** | **-15** |
| **HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH** | **0** | **-23** |
| **PPCHHPPCHPPPPCHHHHCHHPPHHPPPPHHPPHPP** | **0** | **-28** |
| **CPPCHPPCHPPCPPHHHHHHCCPCHPPCPCHPPHPC** | **0** | **-55** |
| **HCPHPCPHPCHCHPHPPPHPPPHPPPPHPCPHPPPHPHHHCCHCHCHCHH** | **0** | **-63** |
| **HCPHPHPHCHHHHPCCPPHPPPHPPPPCPPPHPPPHPHHHHCHPHPHPHH** | **0** | **-45** |

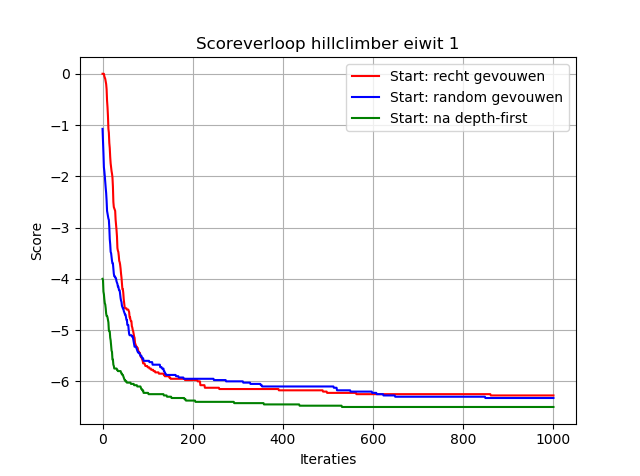
**Experiment 1**

Met dit experiment willen we kijken hoe de beginscore van een streng invloed heeft op de score die uiteindelijk kan worden behaald met het ‘hillclimber’ algoritme. Om deze vraag te beantwoorden, gebruiken we de drie verschillende startpunten waarmee we onze hillclimber kunnen runnen, namelijk:

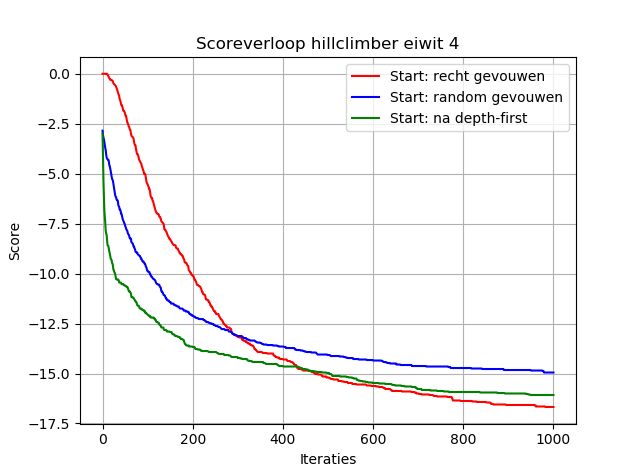
* Straight (recht gevouwen)
* Random (random gevouwen)
* Depth (eerst gerund met depth-first)

Welke runs maken wij:

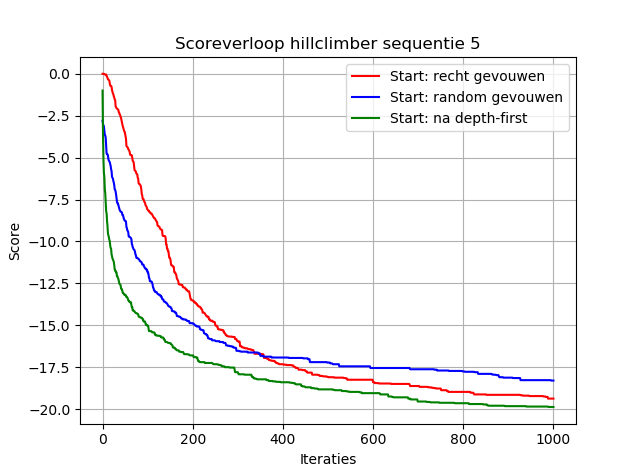
* Elk algoritme runnen wij met de volgende strengs:
  + Streng 1 (alleen P & H, laagste score upper-bound, 14 aminozuren)
  + Streng 4 (alleen P & H, hoogste score upper-bound, 50 aminozuren)
  + Streng 5 (P, H & C, laagste score upper-bound, 36 aminozuren)
  + Streng 7 (P, H & C, hoogste score upper-bound, 50 aminozuren)
* Er is voor deze vier strengs gekozen omdat we graag voor beide soorten strengs (zonder en met cysteïne) het verschil wilden laten zien tussen de streng met de hoogste en de streng met de laagste upper-bound.
* Voor elke combinatie berekenen we de gemiddelde score per iteratie over **40** runs.
* Andere instellingen (voor iedere run gelijk):
  + Dimension: 3D
  + Iteraties : 1000
  + Maximum runtime depth: 15 seconden
* Verwachtingen beginscore:
  + Een recht gevouwen streng heeft altijd een score van 0 omdat de verschillende aminozuren per definitie niet naast elkaar liggen.
  + Een random gevouwen streng heeft altijd een andere score omdat het random is.
  + Een streng vanuit depth-first algoritme heeft altijd dezelfde beginscore per streng. Dit komt omdat we de depth-first eerst 15 seconden laten runnen en dan met de beste streng vanuit daar verder gaan. Omdat hij steeds even lang de tijd krijgt, komt de beste streng altijd neer op dezelfde score.
* Verwachtingen eindscore:
  + We verwachten dat een streng met een lagere beginscore meer en sneller vooruit zal gaan tijdens de hillclimber, maar dat de uiteindelijke score voor alle drie de startpunten ongeveer gelijk zal zijn.
* Resultaten:



**Figuur 1.** Scoreverloop hillclimber sequentie 1. In deze figuur staat op de x-as het iteratie nummer weergegeven, op de y-as staat de gemiddelde score. In deze figuur zijn geplot: de gemiddelde score van 40 runs over het aantal iteraties tijdens de drie verschillende hillclimbers: met startpunt recht gevouwen(rode lijn), met startpunt random gevouwen (blauwe lijn) en na de depth-first search (groene lijn). Alle runs zijn gedaan in een dimensie van 3D en voor het eiwit met sequentie 1.



**Figuur 2.** Scoreverloop hillclimber sequentie 4. In deze figuur staat op de x-as het iteratie nummer weergegeven, op de y-as staat de gemiddelde score. In deze figuur zijn geplot: de gemiddelde score van 40 runs over het aantal iteraties tijdens de drie verschillende hillclimbers: met startpunt recht gevouwen(rode lijn), met startpunt random gevouwen (blauwe lijn) en na de depth-first search (groene lijn). Alle runs zijn gedaan in een dimensie van 3D en voor het eiwit met sequentie 4.



**Figuur 3.** Scoreverloop hillclimber sequentie 5. In deze figuur staat op de x-as het iteratie nummer weergegeven, op de y-as staat de gemiddelde score. In deze figuur zijn geplot: de gemiddelde score van 40 runs over het aantal iteraties tijdens de drie verschillende hillclimbers: met startpunt recht gevouwen(rode lijn), met startpunt random gevouwen (blauwe lijn) en na de depth-first search (groene lijn). Alle runs zijn gedaan in een dimensie van 3D en voor het eiwit met sequentie 5.

**Figuur 4.** Scoreverloop hillclimber sequentie 7. In deze figuur staat op de x-as het iteratie nummer weergegeven, op de y-as staat de gemiddelde score. In deze figuur zijn geplot: de gemiddelde score van 40 runs over het aantal iteraties tijdens de drie verschillende hillclimbers: met startpunt recht gevouwen(rode lijn), met startpunt random gevouwen (blauwe lijn) en na de depth-first search (groene lijn). Alle runs zijn gedaan in een dimensie van 3D en voor het eiwit met sequentie 7.

* Conclusie
  + Wij zien dat vooral de hillclimber met startpunt random een wat slechtere score behaalt dan de andere twee hillclimbers. Onze hypothese hiervoor is dat, wanneer de hillclimber probeert een deel van de keten van plek te veranderen door een rotatie, dit bij het random startpunt heel vaak niet kan omdat de keten over elkaar komt te liggen. Daardoor zal de hillclimber waarschijnlijk veel dezelfde veranderingen teweegbrengen (heen en weer roteren en veel aan het begin en aan het einde van de keten ipv in het midden). Wij denken dat dit bij startpunt rechtgevouwen niet het geval is, omdat aangezien de keten nog helemaal recht ligt, er vooral aan het begin nog heel veel rotaties mogelijk zijn. Ook bij startpunt na depth-first is dit waarschijnlijk niet het geval, omdat aangezien de keten na 15 seconden vouwen gewoon gestopt is met verder gaan, alleen het begin van de keten is gevouwen en de rest nog helemaal recht ligt. In het begindeel van de keten zijn dus nog weinig rotaties mogelijk, maar wij denken dat dit maar een klein deel van de keten is.
  + Om deze hypothese te testen stellen wij de volgende twee onderzoeksvragen:
    1. Hoeveel errors vinden er plaats per iteratie in de drie verschillende hillclimbers?
    2. Hoe vaak wordt welke verbinding in de keten gevouwen?
  + Onze hypotheses voor deze onderzoeksvragen zijn als volgt:
    1. Wij denken dat bij de hillclimber met starpunt random (vooral tijdens de eerste paar iteraties) het aantal errors per iteratie hoger zal zijn dan bij de hillclimbers met startpunt na depth-first en startpunt recht gevouwen.
    2. Wij denken dat bij de hillclimber met startpunt random de verbindingen aan de buitenkant van de keten vaker zullen worden gevouwen (dus bij een keten van 14 lang, bijvoorbeeld vooral verbindingen 0, 1, 2, 11, 12 en 13)
  + Wij zullend de volgende runs maken om onze hypotheses te testen:
    1. Omdat bij sequentie 4 en sequentie 7 in 3d het verschil wat wij testen het duidelijkst aanwezig was, zullen wij testen met deze twee sequenties in 3d.
       - Randomhillclimber, sequentie 4, 3d, 1000 iteraties, gemiddelde van 40 keer