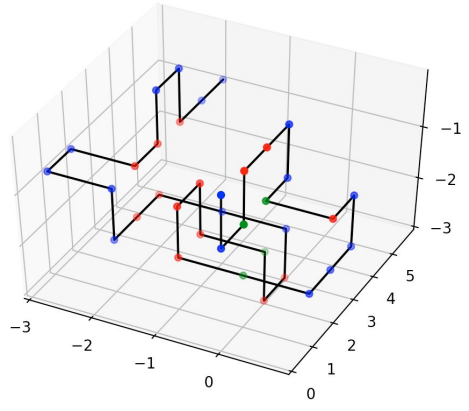
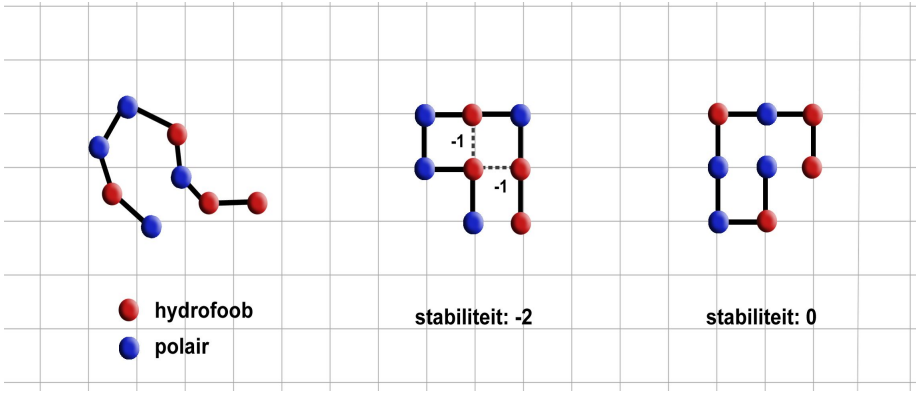
The Triforce of Pow(d)er

Protein folding algorithms

Goal of the case

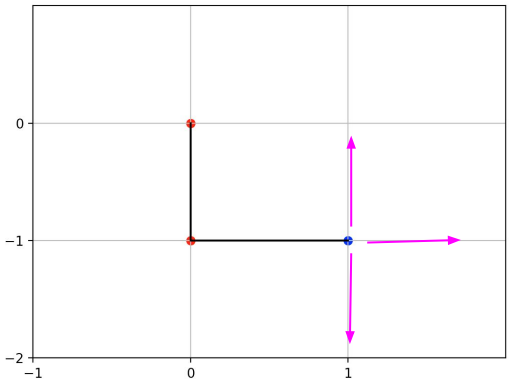
- To fold a given protein as stable as possible in a 2D (or advanced 3D) lattice.
- Stability of a protein depends on its H-/C-bonds.
- The more H-/C-bonds, the more stable the protein is.

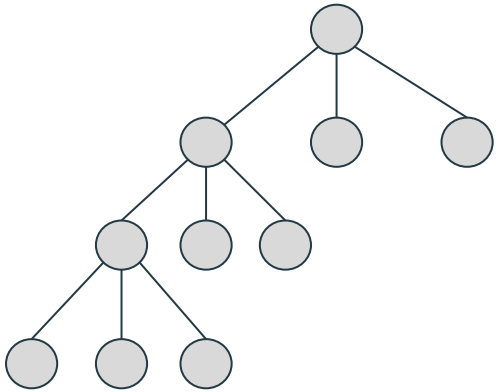




State space (2D)

- Every amino (except the first one) has three possible folding directions
- Upper bound: 3^n
- Smaller due to self-avoiding constraint (complex calculation)





Methods

1. Random
2. Greedy
3. Hillclimber
4. Simulated Annealing
5. Breadth First Search ++++
6. Branch & Bound

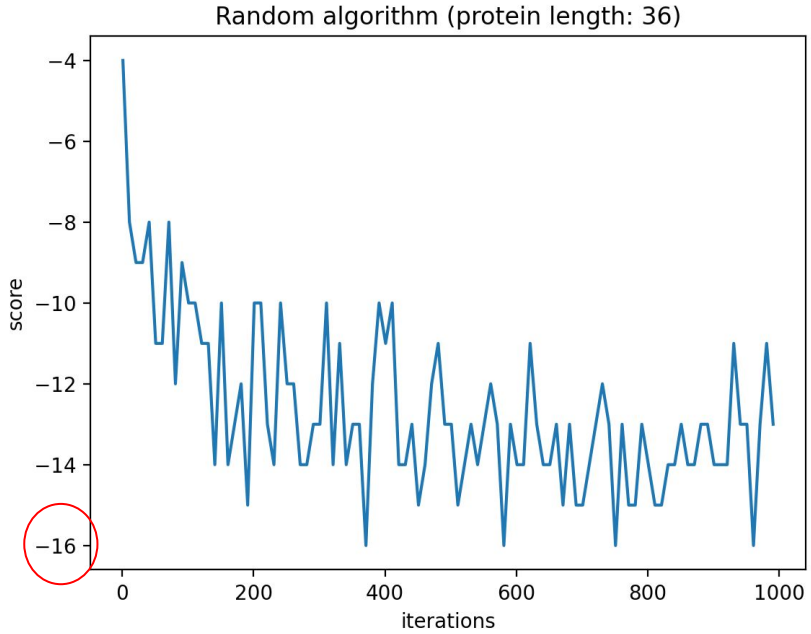
Random

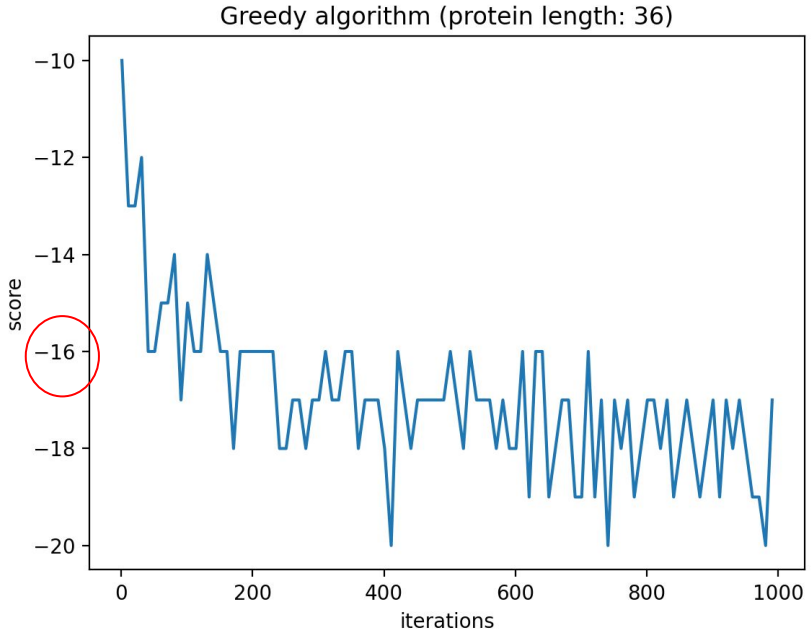
- Folds protein for every iteration by randomly assigning possible values to every amino one by one. Always saves the best final configuration.
- Starts over if folding results in dead end

Greedy

- Architecture: Greedy(RandomFolder)
- Instead of choosing random possible values, the algorithm now chooses (randomly one of) the best values, given the configuration at that point.

Random VS Greedy





Better results with less iterations!

Hillclimber

- Mutates a protein *i* times by changing *n* random amino folds.
- Each improvement is kept for the next iteration

```
for every iteration:

    until all mutations per iteration succeeded:
        mutate protein
         if mutated protein is not valid:
            undo mutation

    if score < best score:
        update beste score
    else:
        undo mutation series
```

Hillclimber: mutations per iteration

4 mutations per iteration seems to be a reasonable choice.

Simulated Annealing

- Architecture: SimulatedAnnealing(HillClimber)
- Difference: sometimes accepts mutated configurations that are worse, depending on the current temperature

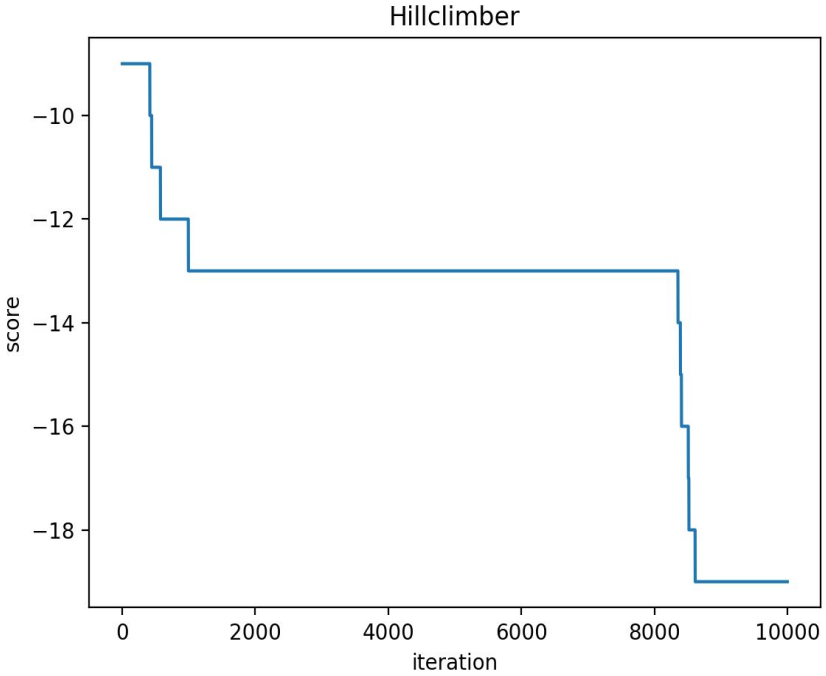
for every iteration:

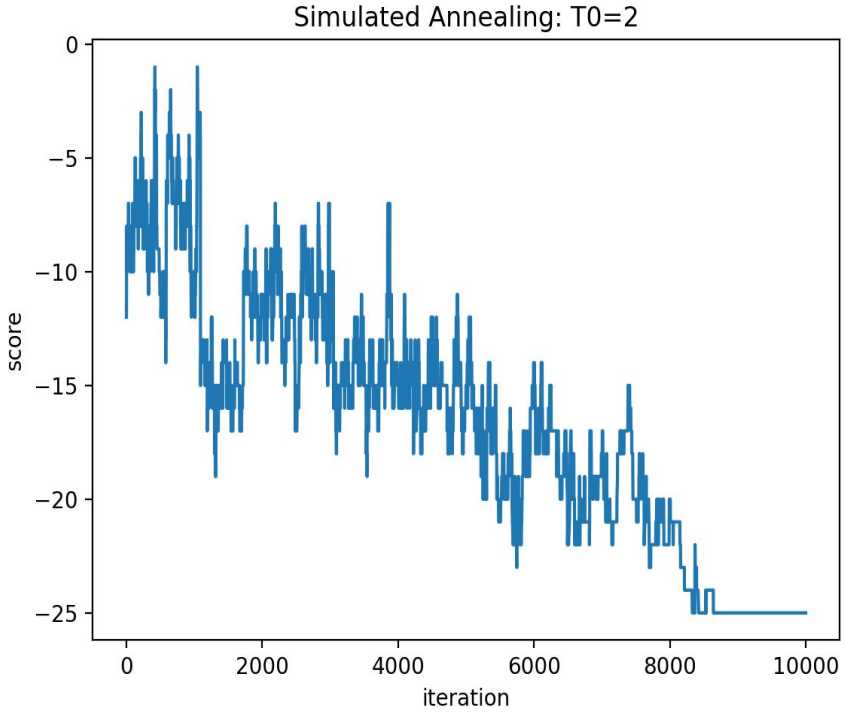
Hillclimber routine

probability = $e^{(\text{best score} - \text{score})/\text{temperature}}$
if random probability < probability:
    update beste score
else:
    undo mutation series

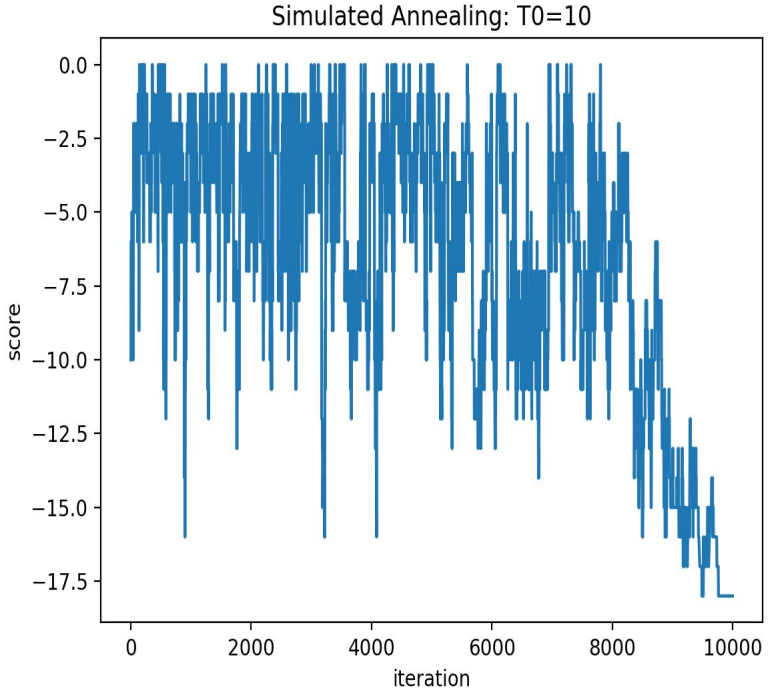
update temperature (linear cooling)

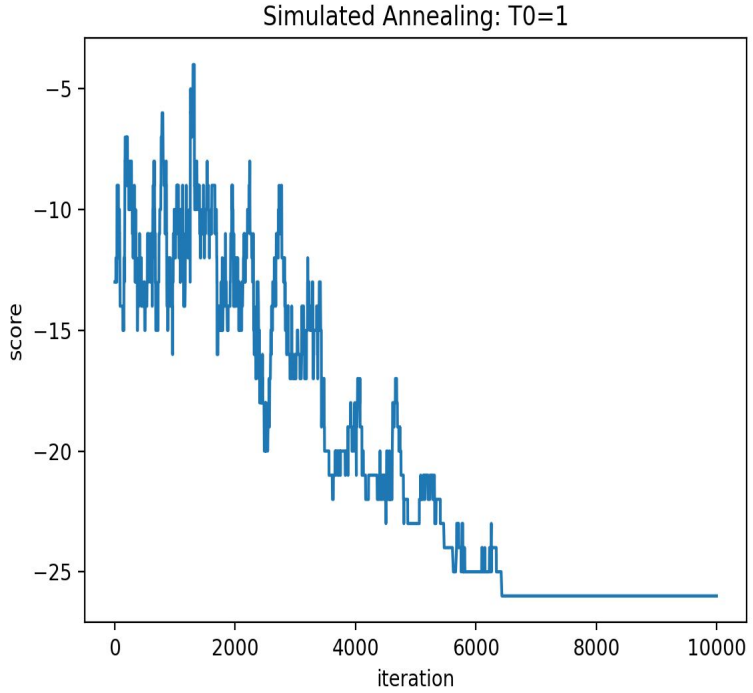
Hillclimber VS Simulated Annealing





Simulated Annealing: starting temperature

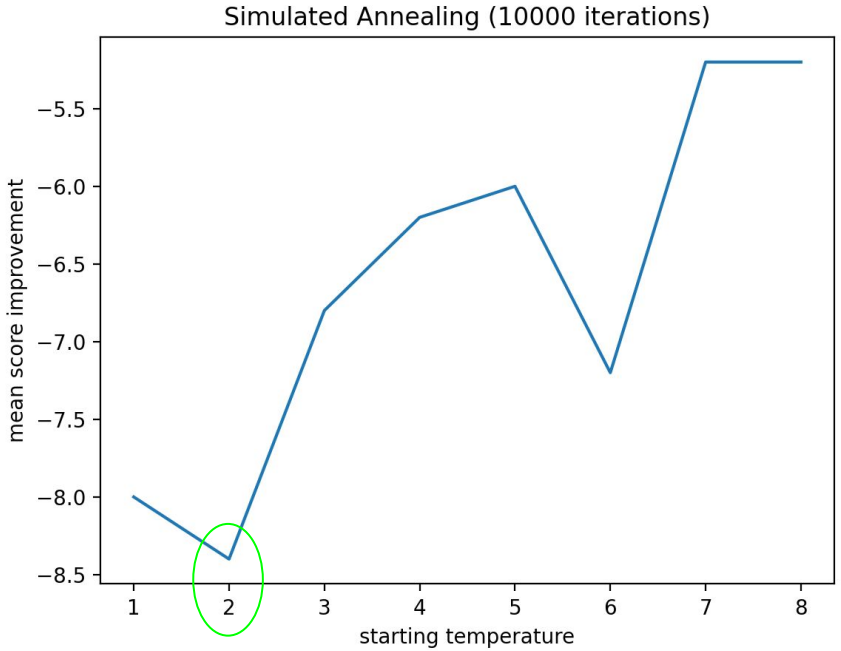




Temperature of 10 accepts too long and too much higher scores.

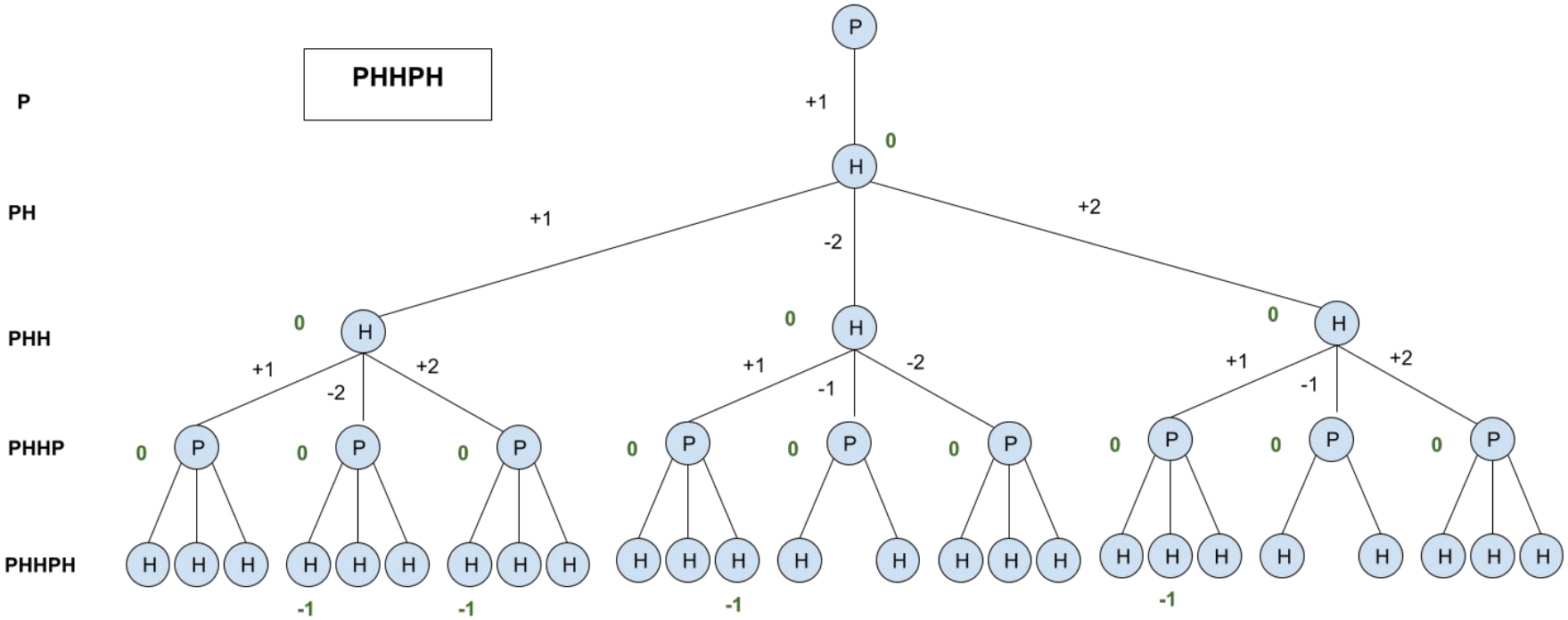
Temperature of 1 rejects higher scores quite early.

Simulated Annealing: starting temperature

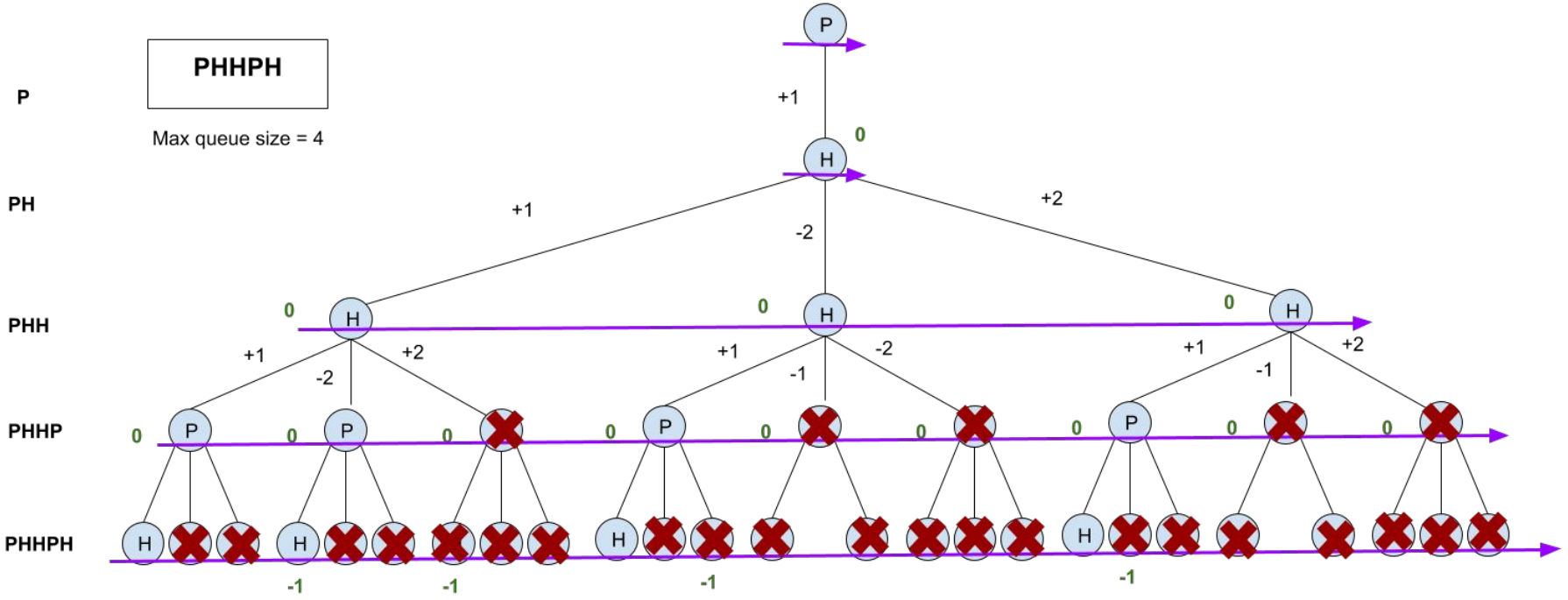


Plot confirms optimal starting temperature of 2

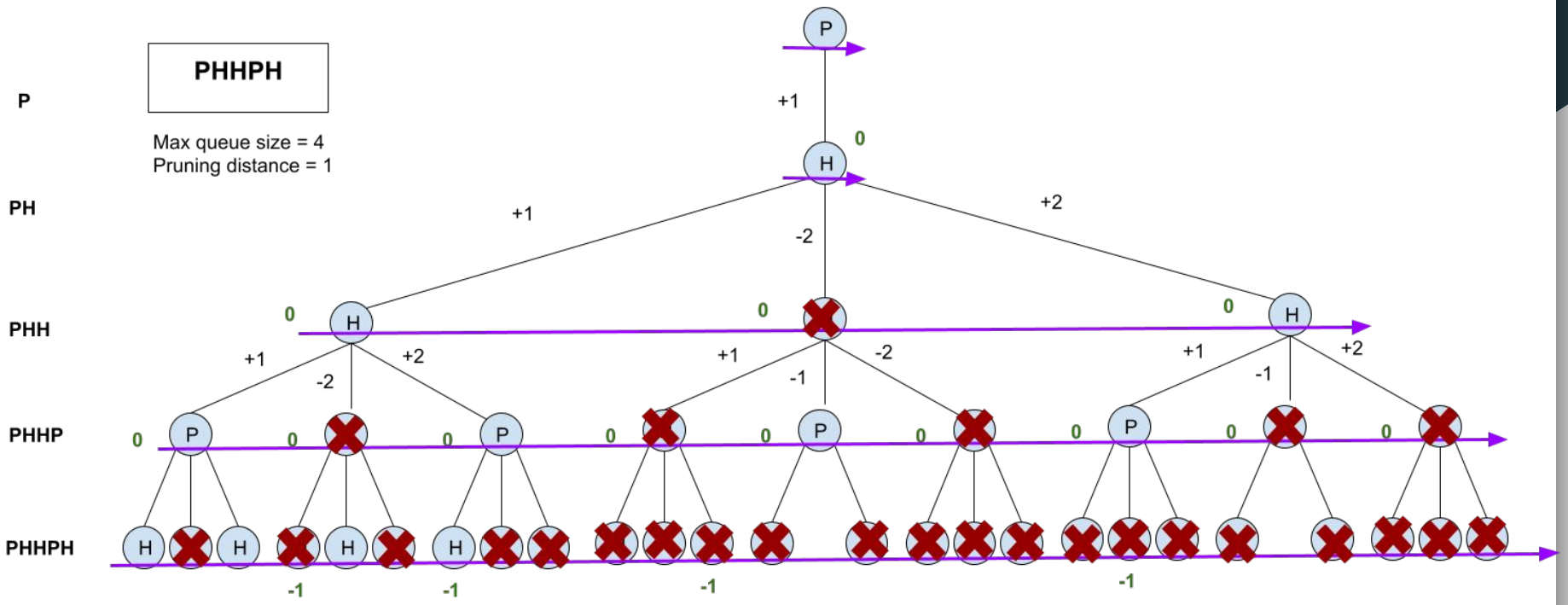
Breadth First Search



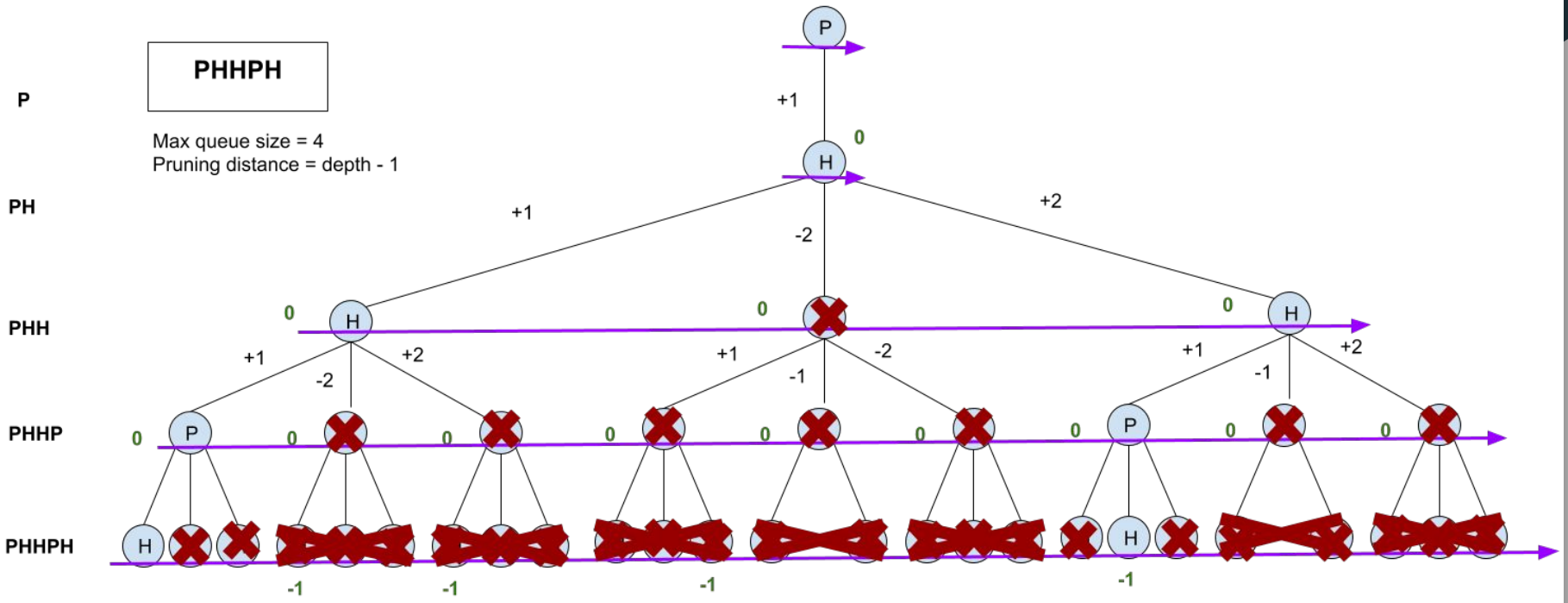
Breadth First search + Max queue size



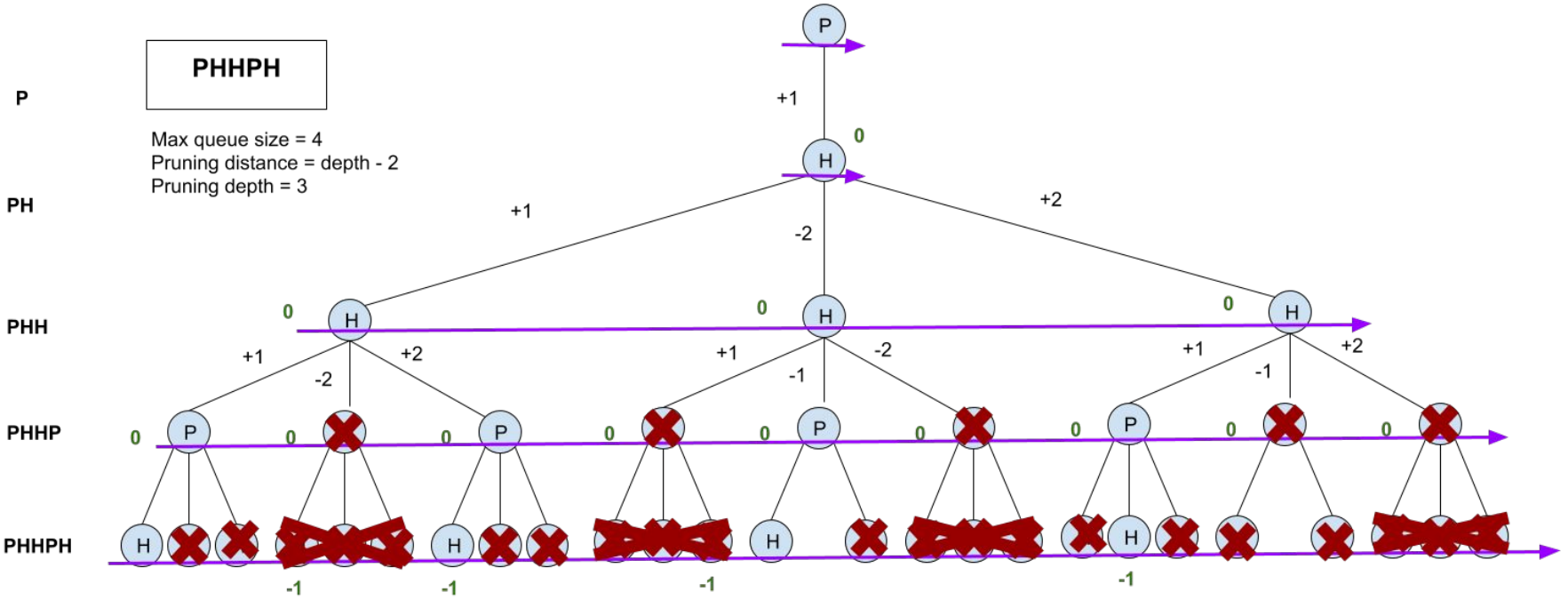
BFS + Max queue size + Pruning distance



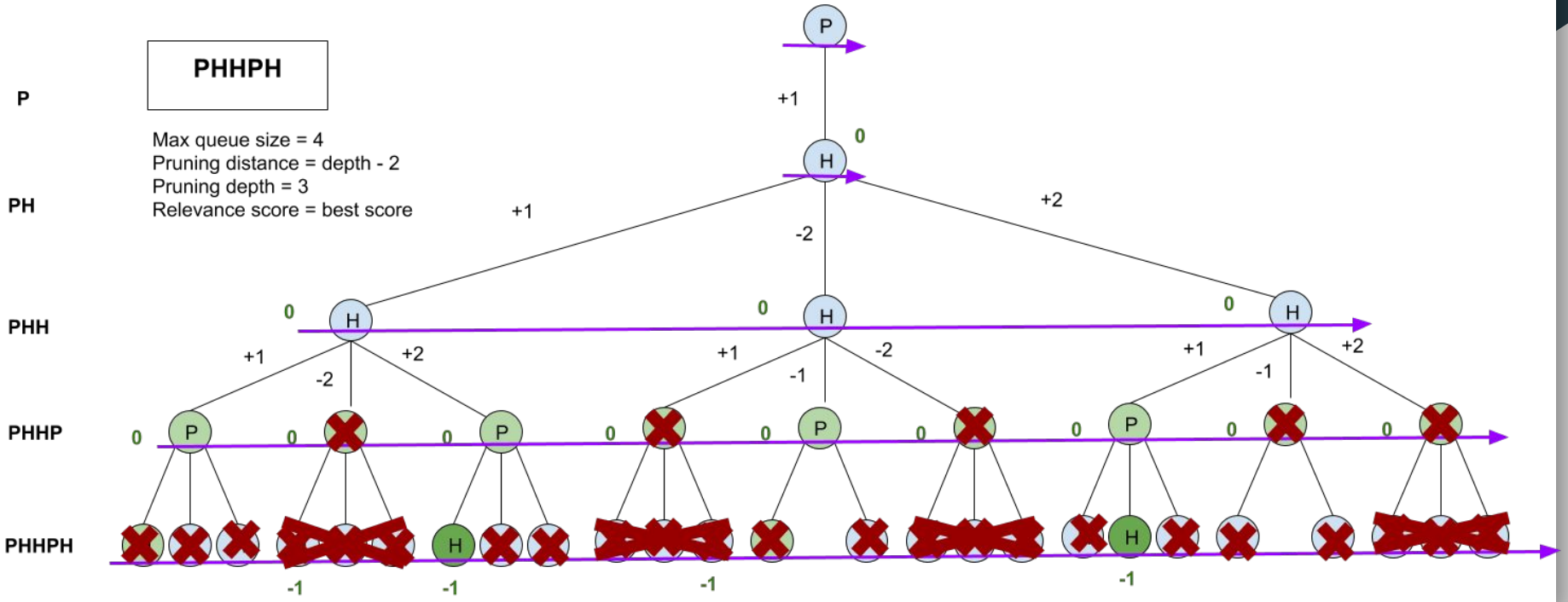
BFS + Max queue size + Pruning distance



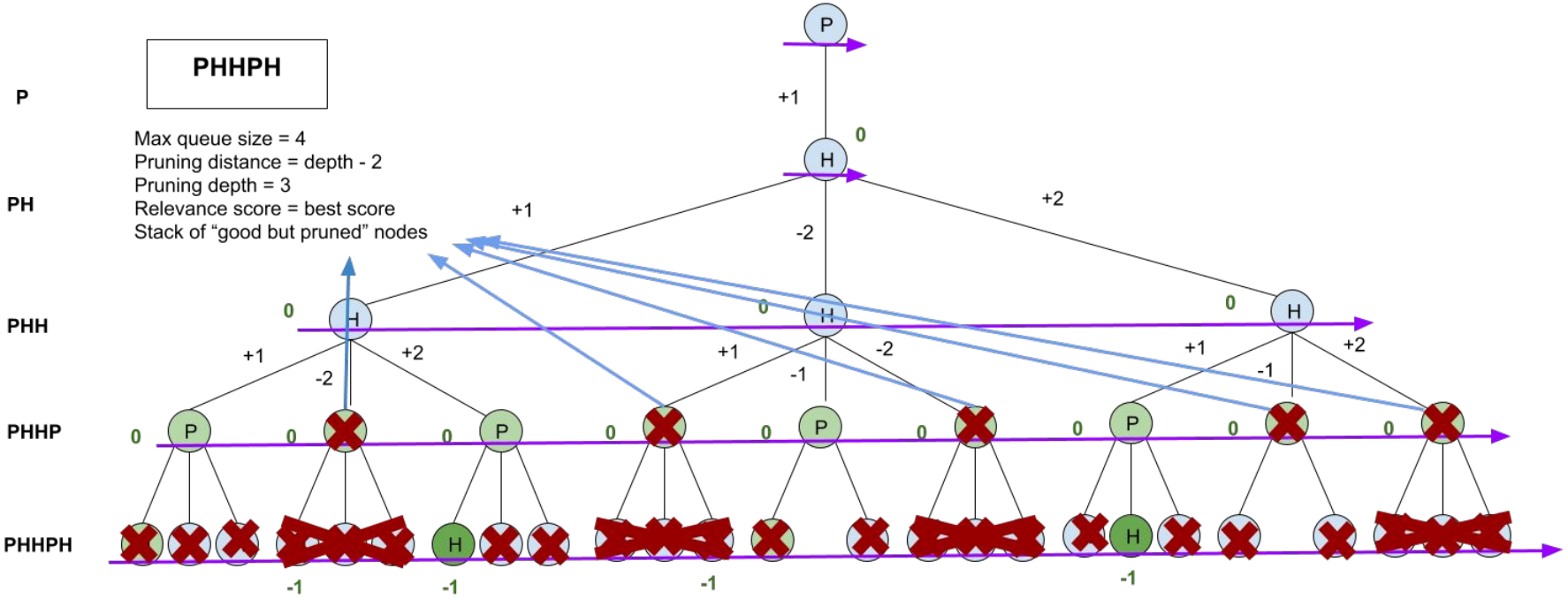
BFS + Max queue size + Pruning distance + Pruning depth



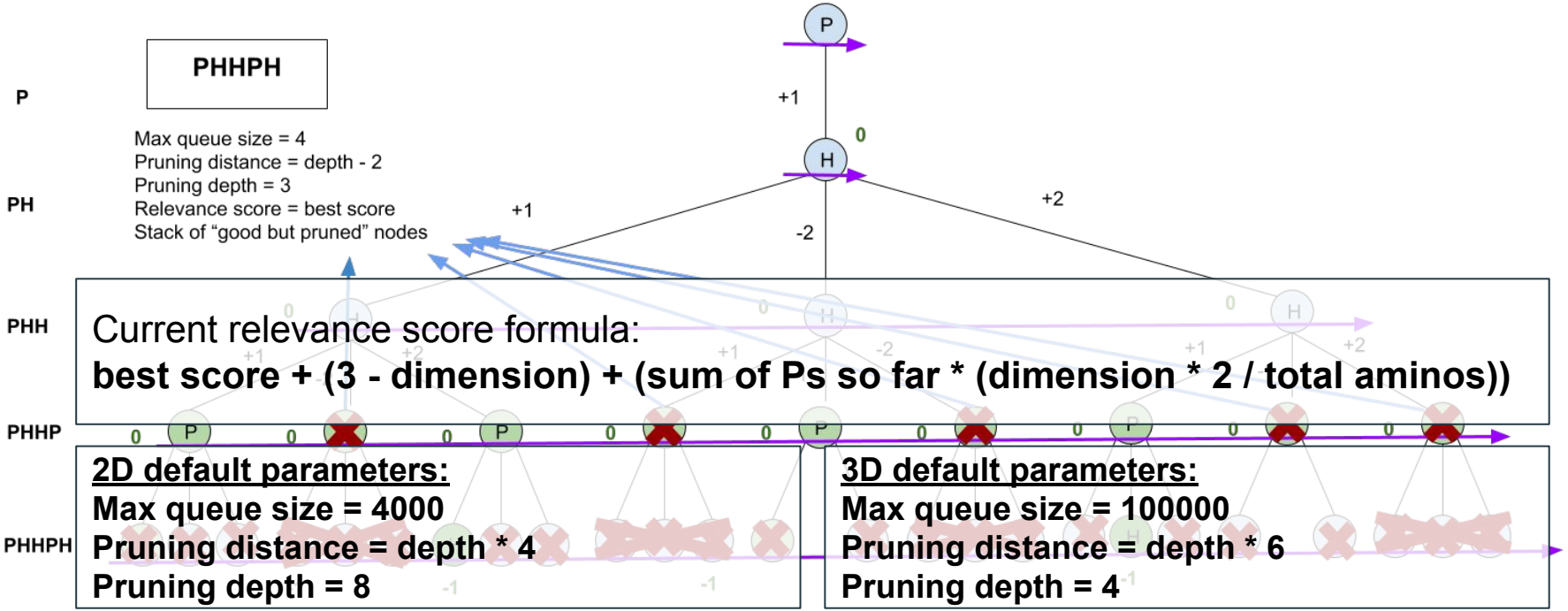
BFS + Max queue size + Pruning distance + Pruning depth + Relevance pruning



BFS + Max queue size + Pruning distance + depth + Relevance + Good nodes archive



BFS + Max queue size + Pruning distance + depth + Relevance + Good nodes archive

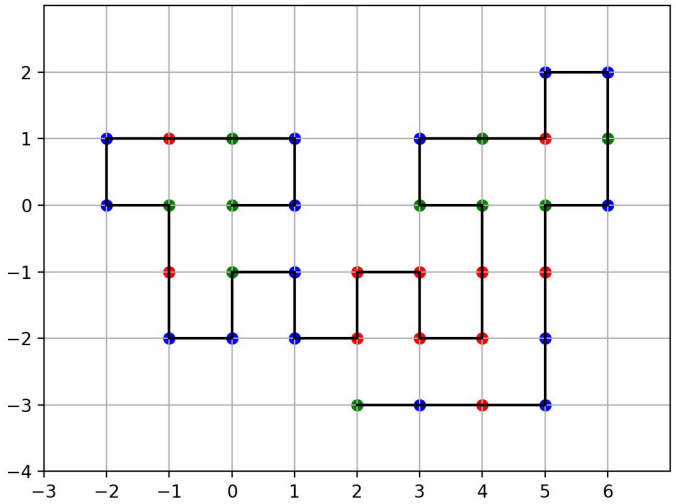


Branch & Bound

- Constructive recursive depth first search
- Pseudocode of subroutine ⟶
- p1 and p2 are resp. 0.8 and 0.5
- Based on: Chen, M., & Huang, W. Q. (2005). A branch and bound algorithm for the protein folding problem in the HP lattice model. Genomics, proteomics & bioinformatics, 3(4), 225-230.
- Unfortunately, the runtime of our implementation makes the algorithm unsuitable for proteins longer than 20 amino acids or 3D.

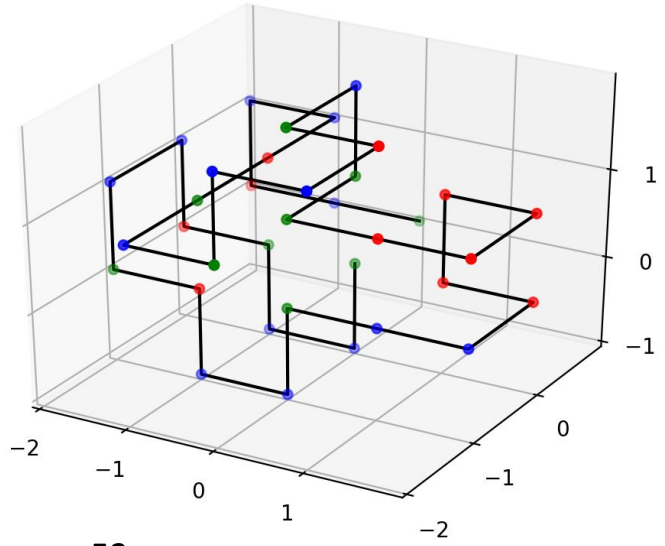
```
search(id):

    get possible values(amino)
    for values in possible values:
        set values
        calculate score
        ...
        if amino is 'H' or 'C':
            if score <= best score[id]:
                search(id + 1)
            elif score > mean score[id]:
                if r > p1:
                    search(id + 1)
            elif score > best score and <= mean score:
                if r > p2:
                    search(id + 1)
            else:
                reset values of last folding
        else:
            search(id + 1)
```

2D



Score: -35

3D



Score: -52

Protein with length 36 with Cysteine aminos (CPPCHPPCHPPCPPHHHHHHHCCPCHPPCPCHPPHPC)

Results

Protein folding scores 2D without C (around 5 min. runtime)

Protein length	Random	Greedy	HillClimber	Simulated Annealing	BFS+	Branch & Bound
14	-6	-6	-6	-6	-6	-6
20	-8	-9	-8	-9	-9	-8
36	-9	-13	-10	-12	-13	-12 took 20 minutes.
50	-13	-15	-16	-19	-18	-

Results

Protein folding scores 2D with C (around 5 min. runtime)

Protein length	Random	Greedy	HillClimber	Simulated Annealing	BFS+
36	-17	-19	-20	-24	-21
36	-29	-35	-33	-35	-33
50	-23	-26	-26	-28	-26
50	-21	-30	-29	-28	-29

Results

Protein folding scores 3D without C (around 5 min. runtime)

Protein length	Random	Greedy	HillClimber	Simulated Annealing	BFS+
14	-7	-7	-7	-7	-7
20	-9	-11	-11	-11	-10
36	-11	-16	-16	-18	-16
50	-13	-25	-26	-24	-27

Results

Protein folding scores 3D with C (around 5 min. runtime)

Protein length	Random	Greedy	HillClimber	Simulated Annealing	BFS+
36	-19	-27	-28	-34	-22
36	-33	-54	-54	-51	-44
50	-25	-39	-42	-40	-39
50	-20	-41	-41	-41	-37

Conclusions

- No single best algorithm
- The choice of an algorithm strongly depends on the protein
- The random algorithm rarely seems to be a good choice
- BFS does not excel in proteins with Cysteine.

Discussion & future plans

- A lot of randomness (impact on parameter selection)
- Taking protein length and composition into account for algorithm implementation and parameter selection
- Improving Branch and Bound runtime:
  - experimenting with different probabilities
  - maybe hashed proteins that take little memory and time
- Visualisations of algorithm running process: dynamic tree-visualisations, ...