Using Quantum Annealing to Fold and Design HP Lattice Proteins

Lucas Knuthson[1]
Centre for Environmental and Climate Science (CEC), Lund University
Nordic Lattice 2024 in Lund

With: Anders Irbäck, Sandipan Mohanty (Jülich), Carsten Peterson

[1]lucas.knuthson@cec.lu.se

Introduction

- Quantum Annealing (D-Wave)
  - QPU
  - Hybrid
- HP Lattice Proteins
- Folding **[Irbäck et al. 2022, PhysRevResearch.4.043013]**
  - Mapping and Energy Function
  - Results: Hybrid
  - Results: QPU
  - Conclusion
- Design **[Irbäck et al. 2024, PhysRevResearch.6.013162]**
  - Mapping and Energy Function
  - Results: Hybrid
  - Results: QPU
  - Conclusion
- If I have time: Some QAOA

Quantum Computing

Replace bits $\{0,1\}^n$ with qubits $|state\rangle = \bigotimes_i^n (\alpha_i |0\rangle + \beta_i |1\rangle)$

Quantum Computing

Replace bits $\{0,1\}^n$ with qubits $|state\rangle = \bigotimes_i^n (\alpha_i |0\rangle + \beta_i |1\rangle)$

Main effort is to make gate-based quantum computers
$|f\rangle = U_n \cdots U_1 |start\rangle$

Quantum Computing

Replace bits $\{0,1\}^n$ with qubits $|state\rangle = \bigotimes_i^n (\alpha_i |0\rangle + \beta_i |1\rangle)$

Main effort is to make gate-based quantum computers
$|f\rangle = U_n \cdots U_1 |start\rangle$

Quantum annealing: $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) |start\rangle$,
$H(t) = H_P(t/T) + H_D(1 - t/T)$

H_p is the problem Hamiltonian, and H_D driver in which ground state we start in

Quantum Computing

Replace bits $\{0,1\}^n$ with qubits $|state\rangle = \bigotimes_i^n (\alpha_i |0\rangle + \beta_i |1\rangle)$

Main effort is to make gate-based quantum computers
$|f\rangle = U_n \cdots U_1 |start\rangle$

Quantum annealing: $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) |start\rangle$,
$H(t) = H_P(t/T) + H_D(1 - t/T)$

H_p is the problem Hamiltonian, and H_D driver in which ground state we start in

Guaranteed to find the ground state if $T >> \max \frac{\langle gs(t)|\dot{H}|e(t)\rangle}{(E_{gs}(t) - E_e(t))^2}$

Quantum Annealing: D-Wave

Want to solve optimization problems, $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) \, |start\rangle$.

Quantum Annealing: D-Wave

Want to solve optimization problems, $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) |start\rangle$.

We wish to find the ground state of $H_P = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$

Quantum Annealing: D-Wave

Want to solve optimization problems, $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) |start\rangle$.

We wish to find the ground state of $H_P = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$

Start in the ground state, $|+\rangle^n$, of $H_D = -\sum_i \sigma_i^x$

Quantum Annealing: D-Wave Advantage

Newest Machine D-Wave Advantage has 5000+ qubits and 15-way connectivity

Quantum Annealing: D-Wave Advantage

Newest Machine D-Wave Advantage has 5000+ qubits and 15-way connectivity

Pure QPU solver: If the connectivity is higher than 15, we create **chains** of physical qubits representing a logical bit

Quantum Annealing: D-Wave Advantage

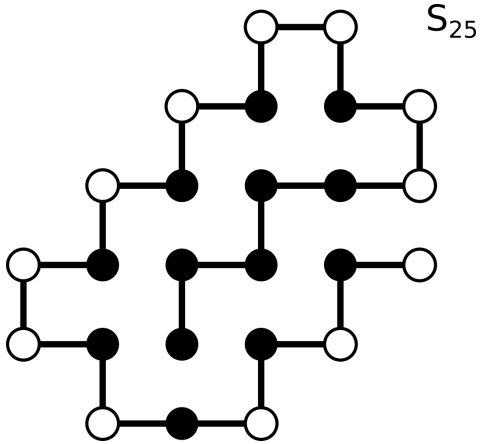
Newest Machine D-Wave Advantage has 5000+ qubits and 15-way connectivity

Pure QPU solver: If the connectivity is higher than 15, we create **chains** of physical qubits representing a logical bit

Hybrid quantum-classical solver: Subproblems sent as queries to the QPU

HP Lattice Proteins [Lau,Dill 1989]

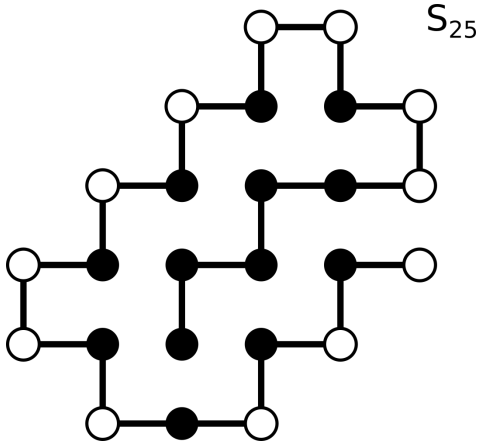
Simplified protein model with two types of amino acids, H and P.
Hs interact, and Ps do not. $E_{\mathrm{HP}} = -N_{\mathrm{HH}}$



HP Lattice Proteins [Lau,Dill 1989]

Simplified protein model with two types of amino acids, H and P. Hs interact, and Ps do not. $E_{\mathrm{HP}} = -N_{\mathrm{HH}}$

For 2D lattice, all sequences with unique ground state structures are known for chains with ≤ 30 amino acids

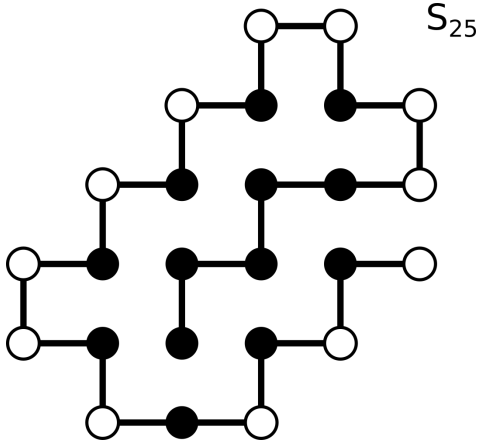


HP Lattice Proteins [Lau,Dill 1989]

Simplified protein model with two types of amino acids, H and P. Hs interact, and Ps do not. $E_{\mathrm{HP}} = -N_{\mathrm{HH}}$

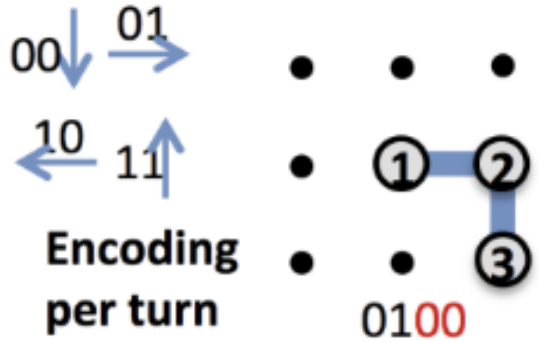
For 2D lattice, all sequences with unique ground state structures are known for chains with ≤ 30 amino acids

Despite the simplicity, both folding (given a sequence find its ground state structure) and design (given a structure find a sequence that folds into that structure) are computationally difficult



Folding HP Lattice Proteins: Questions We Wanted to Answer

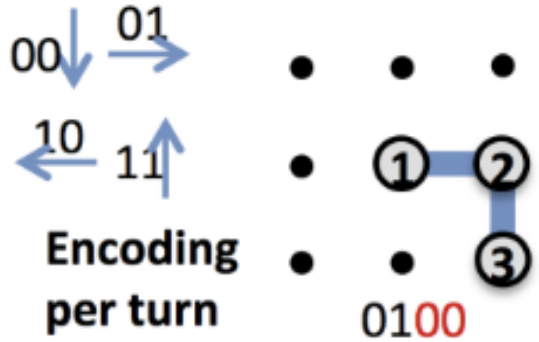
Previous attempts used a turn-based encoding. Non-local interactions became difficult to implement. (short chains 6-9 amino acids)



[Perdomo-Ortiz et al. 2012]

Folding HP Lattice Proteins: Questions We Wanted to Answer

Previous attempts used a turn-based encoding. Non-local interactions became difficult to implement. (short chains 6-9 amino acids)

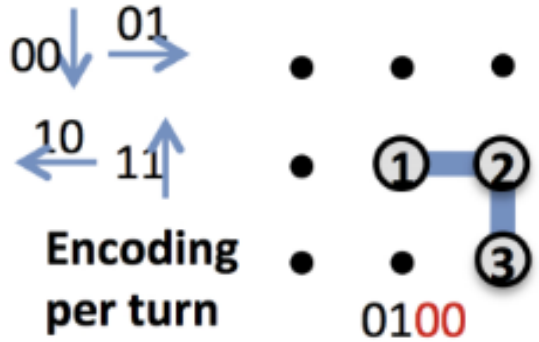


[Perdomo-Ortiz et al. 2012]

Can we find a simpler mapping which is quadratic?

Folding HP Lattice Proteins: Questions We Wanted to Answer

Previous attempts used a turn-based encoding. Non-local interactions became difficult to implement. (short chains 6-9 amino acids)



[Perdomo-Ortiz et al. 2012]

Can we find a simpler mapping which is quadratic?

We wanted to test the mapping on actual hardware

Folding: Mapping and Energy Function

Can we find a simpler mapping which is quadratic?

Folding: Mapping and Energy Function

Can we find a simpler mapping which is quadratic?

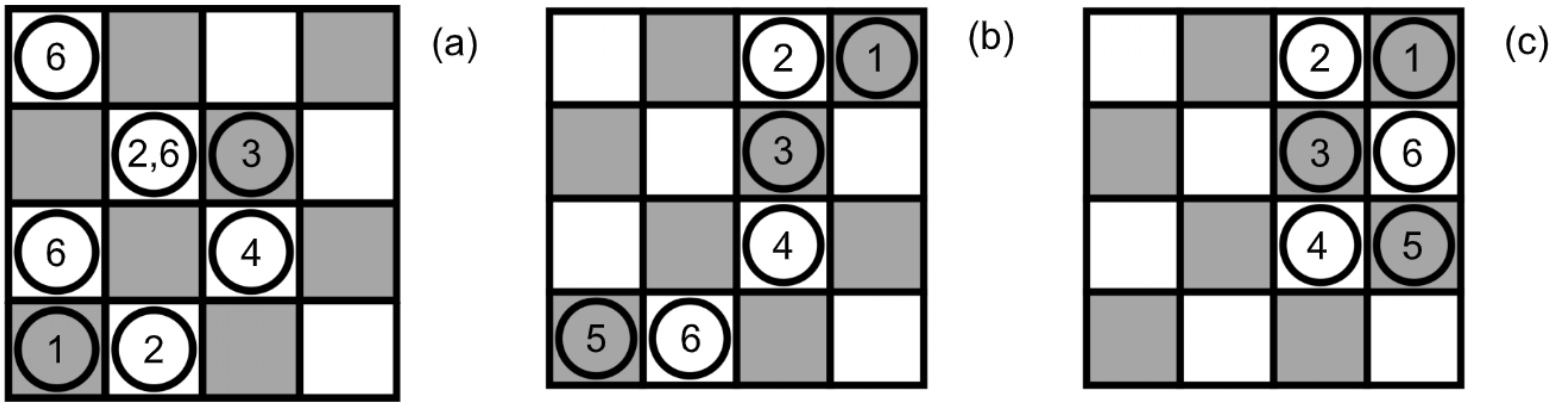
Choose a lattice and enumerate the beads in the sequence

Folding: Mapping and Energy Function

Can we find a simpler mapping which is quadratic?

Choose a lattice and enumerate the beads in the sequence

A spin $\sigma_s^f \in \{0, 1\}$ is 1 if bead f is on site s, $NL^2/2 \approx N^2$ spins



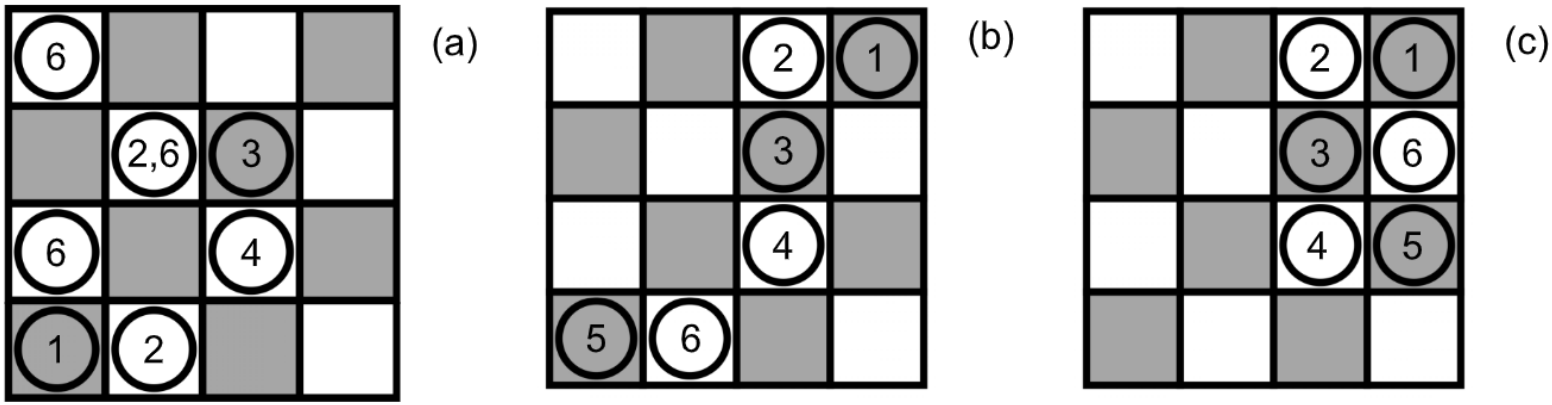
Folding: Mapping and Energy Function

Can we find a simpler mapping which is quadratic?

Choose a lattice and enumerate the beads in the sequence

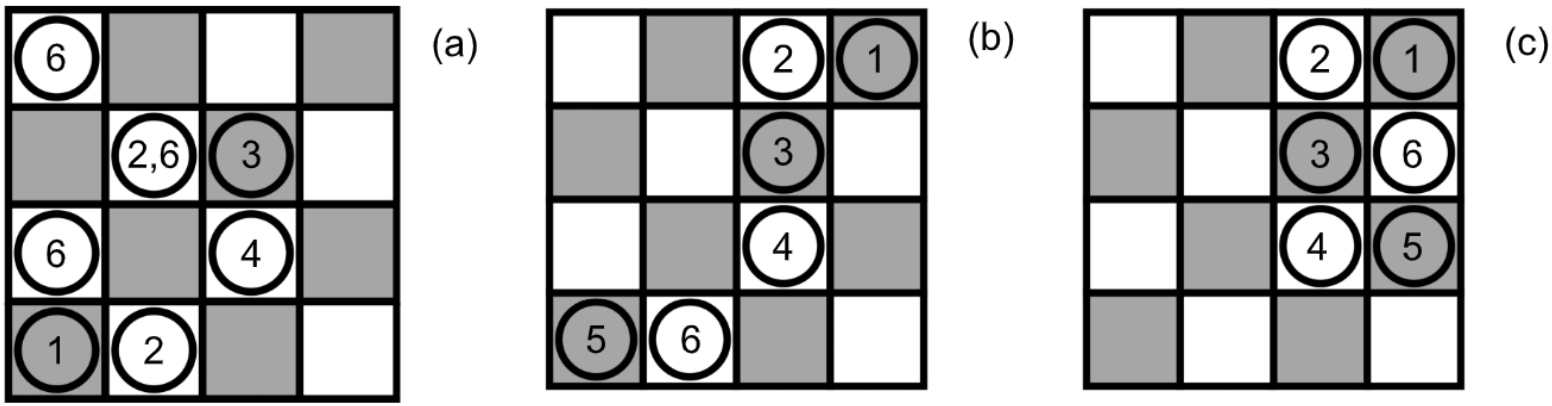
A spin $\sigma_s^f \in \{0, 1\}$ is 1 if bead f is on site s, $NL^2/2 \approx N^2$ spins

Energy function takes the form $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$, with E_i being constraints to ensure a valid chain



Folding: Mapping and Energy Function

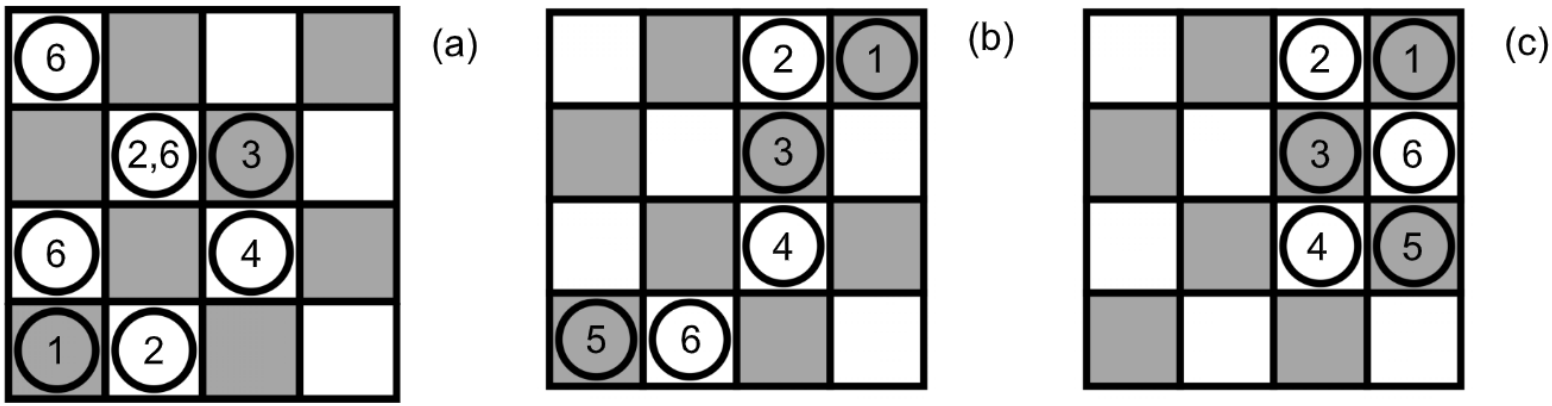
A spin $\sigma_s^f \in \{0, 1\}$ is 1 if bead f is on site s, $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$



Folding: Mapping and Energy Function

A spin $\sigma_s^f \in \{0,1\}$ is 1 if bead f is on site s, $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

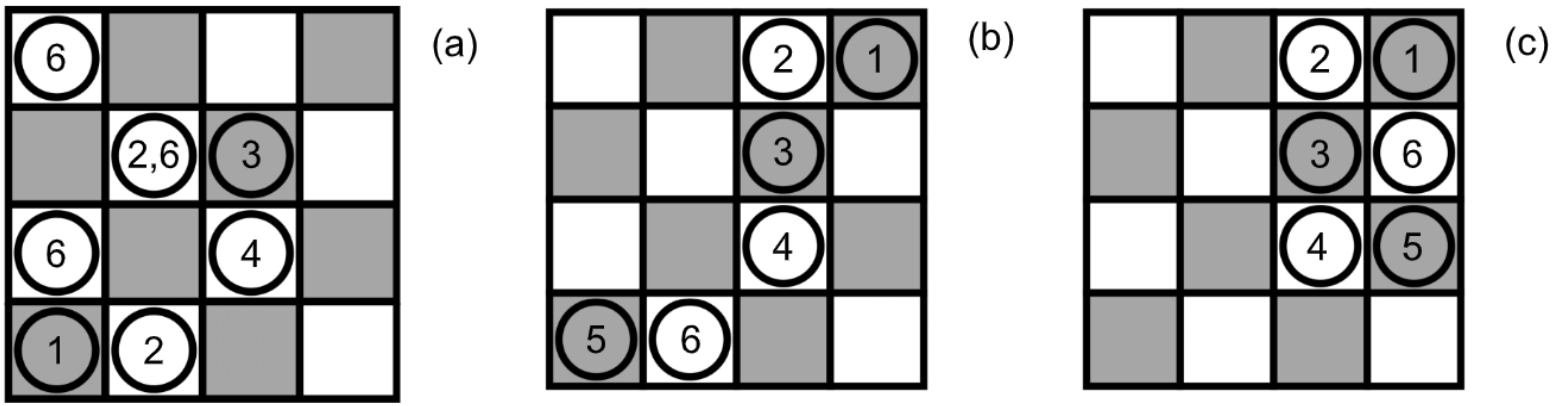


Folding: Mapping and Energy Function

A spin $\sigma_s^f \in \{0,1\}$ is 1 if bead f is on site s, $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2,$$

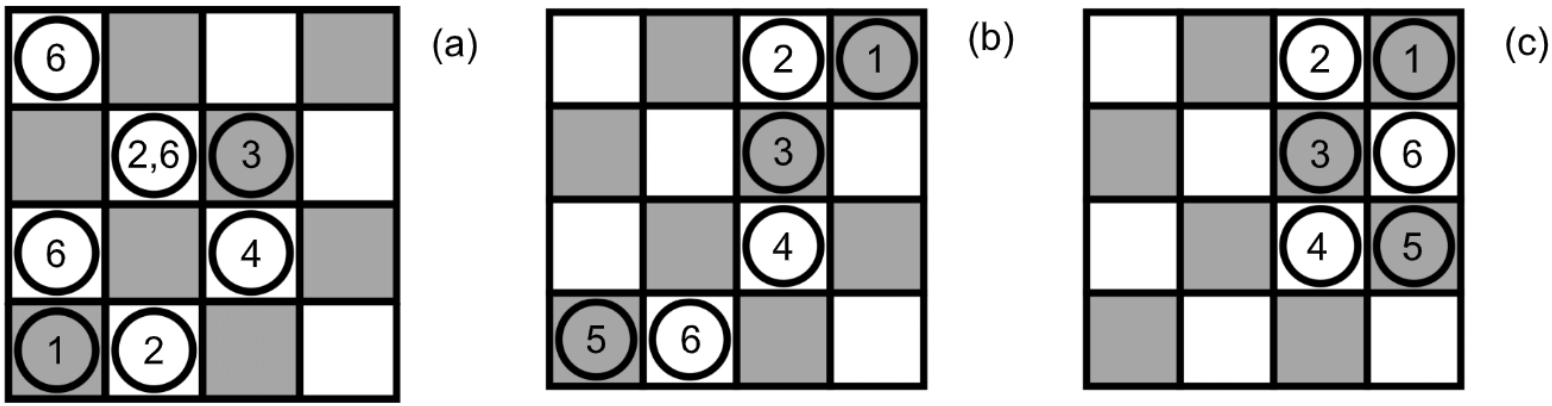


Folding: Mapping and Energy Function

A spin $\sigma_s^f \in \{0,1\}$ is 1 if bead f is on site s, $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$



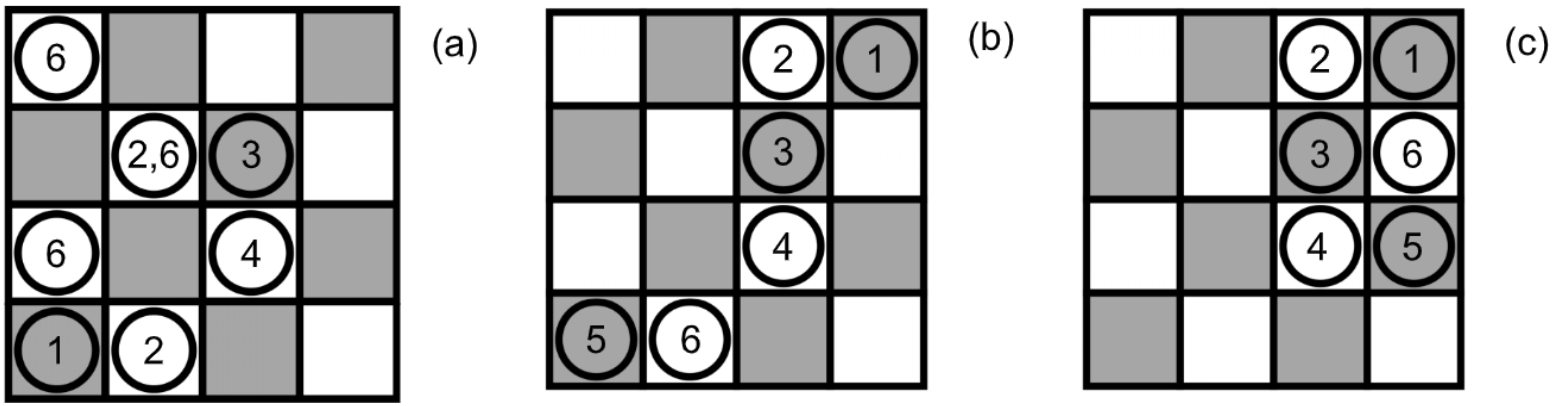
Folding: Mapping and Energy Function

A spin $\sigma_s^f \in \{0,1\}$ is 1 if bead f is on site s, $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$



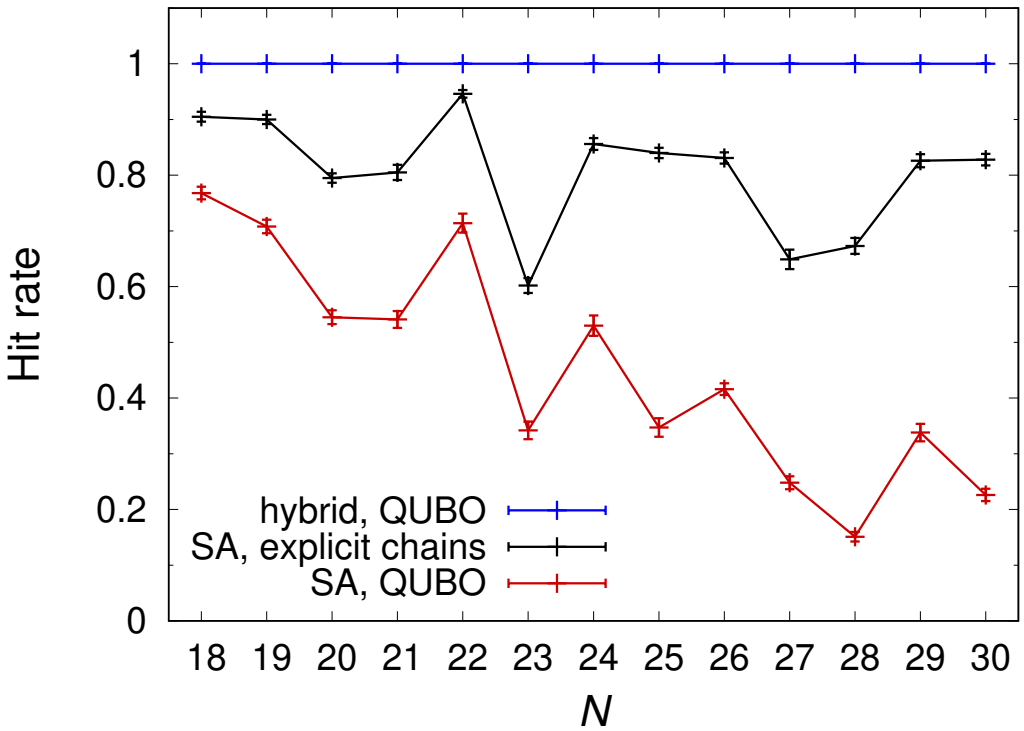
Folding: Hybrid

Implement on hardware: Hybrid minimize $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

Folding: Hybrid

Implement on hardware: Hybrid minimize $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

100 runs for each sequence (10^2 lattice, runtime 4 s) 100% hit rate

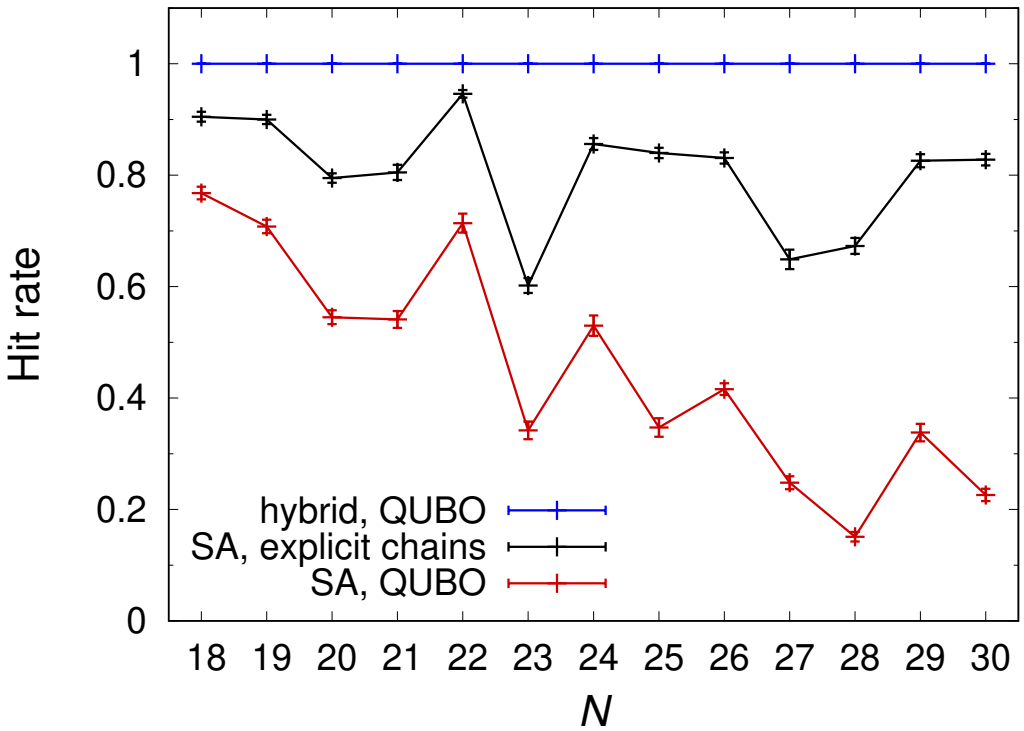


Folding: Hybrid

Implement on hardware: Hybrid minimize $E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$

100 runs for each sequence (10^2 lattice, runtime 4 s) 100% hit rate

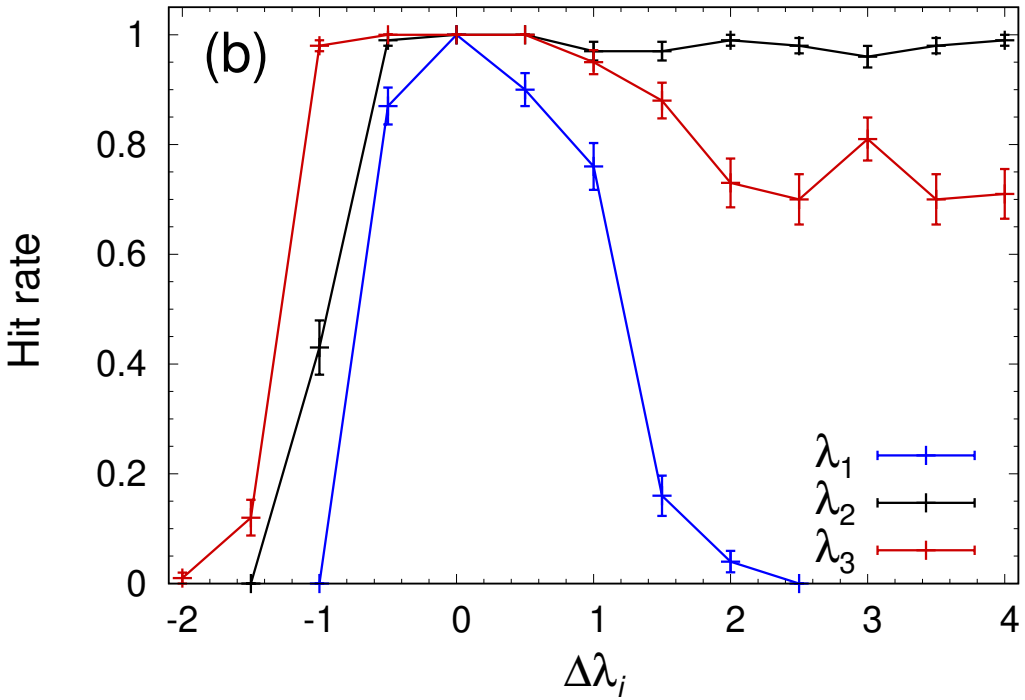
For comparison, classical SA with explicit chains and spins (run time >4 s)



Folding: Hybrid

$$E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i$$

Hybrid solver was insensitive to changes, no fine tuning needed



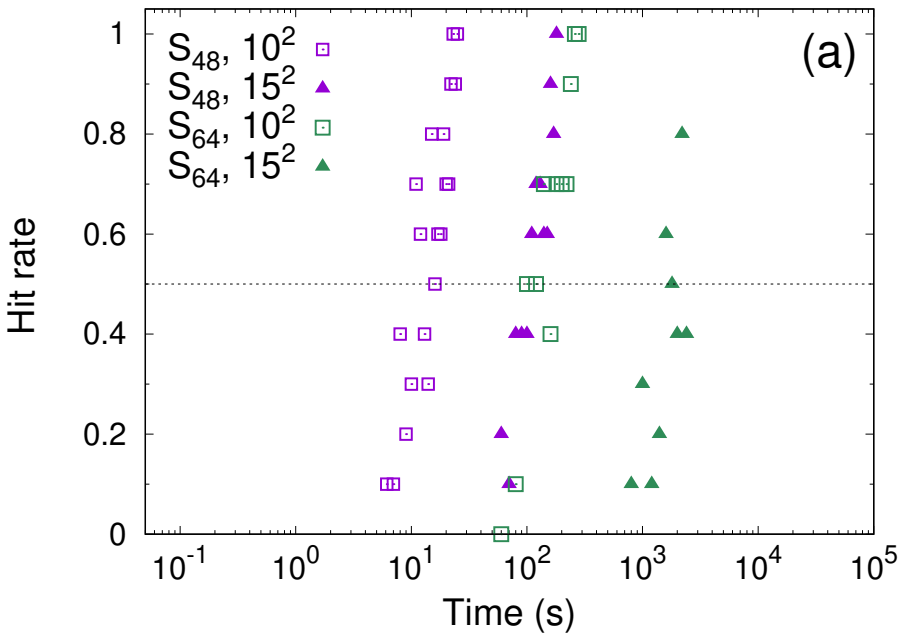
Folding: Hybrid

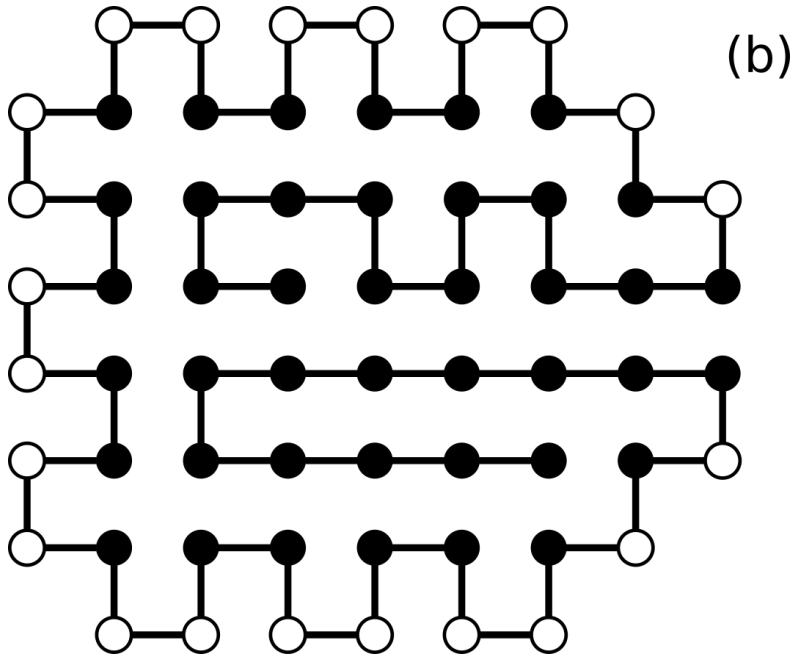
We also tried two longer sequences (without exact results) that have been extensively studied with classical methods

Folding: Hybrid

We also tried two longer sequences (without exact results) that have been extensively studied with classical methods

The lowest known energies were recovered with high probability, once the runtime was high enough





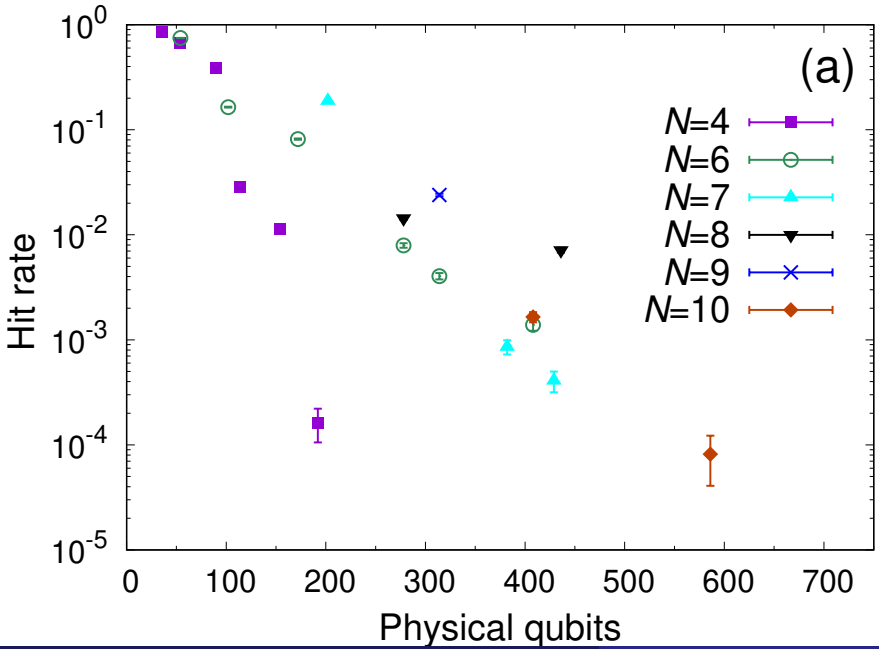
Folding: QPU only

Limited to short sequences

Folding: QPU only

Limited to short sequences

490, 00 attempts per sequence (annealing time 2 ms). Hit rate decays roughly exponentially

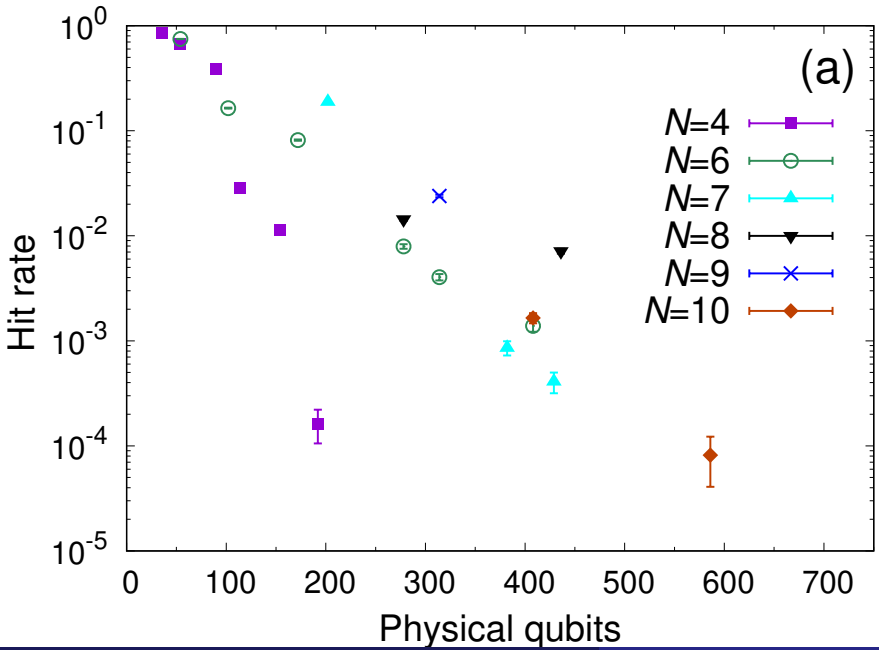


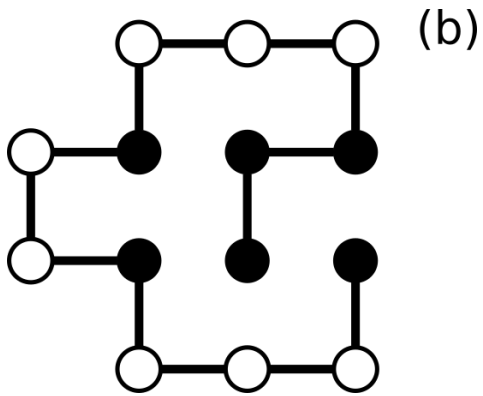
Folding: QPU only

Limited to short sequences

490, 00 attempts per sequence (annealing time 2 ms). Hit rate decays roughly exponentially

Longest sequence was 14-beads on a 4^2 grid





Folding: Conclusion

We found a mapping and energy function that works well for large proteins

Folding: Conclusion

We found a mapping and energy function that works well for large proteins

The hybrid outperforms our SA approaches

Folding: Conclusion

We found a mapping and energy function that works well for large proteins

The hybrid outperforms our SA approaches

The QPU is less impressive, with a roughly exponential decrease in hit rate with increasing system size

Design of HP Lattice Proteins: Questions We Wish to Answer

Can we find a protocol to design HP lattice proteins using QA?

Design of HP Lattice Proteins: Questions We Wish to Answer

Can we find a protocol to design HP lattice proteins using QA?

What is the source for the exponential decay?

Design: Procedure

Goal: find HP sequences that fold to a given target structure. Requires search in both sequence and structure spaces.

Design: Procedure

Goal: find HP sequences that fold to a given target structure. Requires search in both sequence and structure spaces.

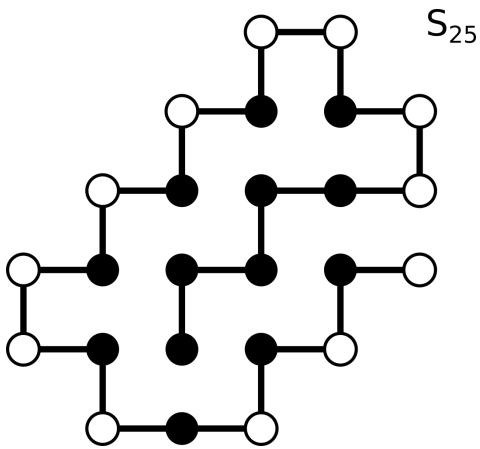
Two-step procedure:

1. Seek sequences with minimum energy in the target structure
2. Test whether or nor the optimized sequence fold to the intended structure (using the folding mapping)

Step 1 is done at fixed number of Hs. Otherwise, the all-H sequence is a trivial solution.

Design: Sequence Optimization

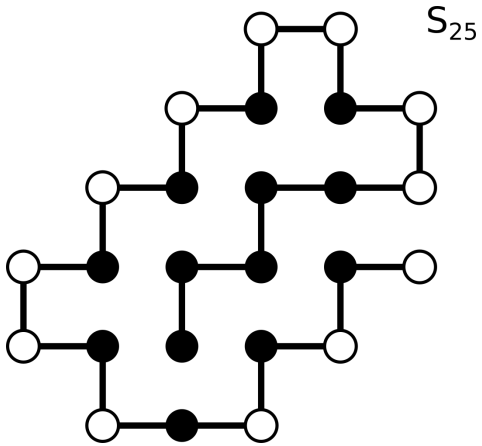
The connectivity matrix of a target structure, w_{ij} tells whether amino acid i and j are in contact ($w_{ij} = -1$) or not ($w_{ij} = 0$)



Design: Sequence Optimization

The connectivity matrix of a target structure, w_{ij} tells whether amino acid i and j are in contact ($w_{ij} = -1$) or not ($w_{ij} = 0$)

s_i indicates if amino acid i is H ($i = 1$) or P ($i = 0$)

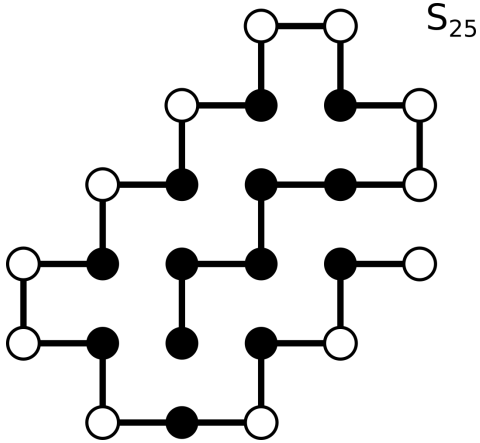


Design: Sequence Optimization

The connectivity matrix of a target structure, w_{ij} tells whether amino acid i and j are in contact ($w_{ij} = -1$) or not ($w_{ij} = 0$)

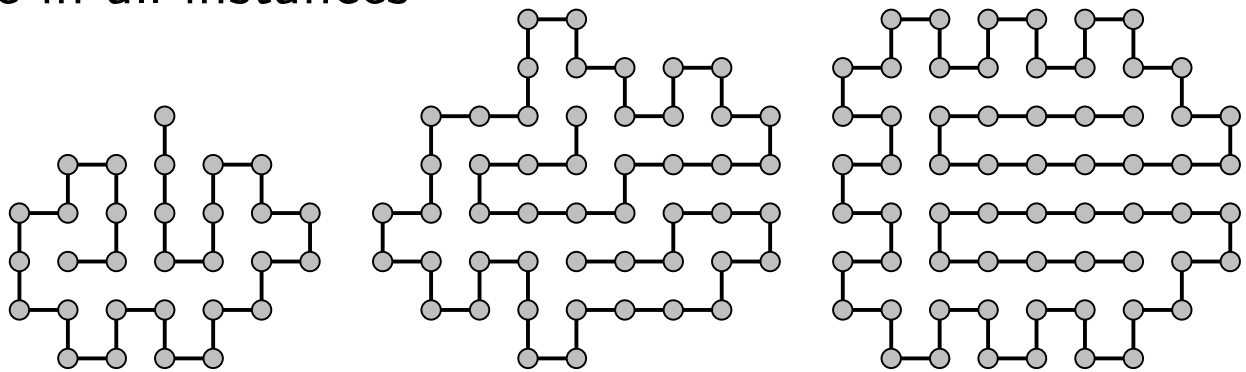
s_i indicates if amino acid i is H ($i = 1$) or P ($i = 0$)

Find minimum-energy sequences for a given composition, N_{H}, by minimizing $E = \sum_{ij} w_{ij} s_i s_j + \lambda \left(\sum_i s_i - N_{\mathrm{H}} \right)^2$



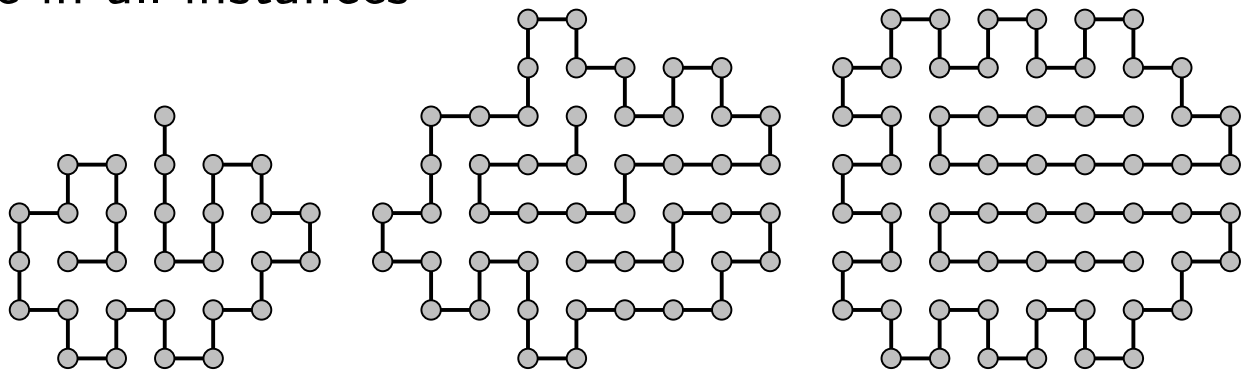
Design: Sequence Optimization with Hybrid

Target structures with $N = 30, 50, 64$.
a few different N_{H} values for each structure
100% success rate in all instances

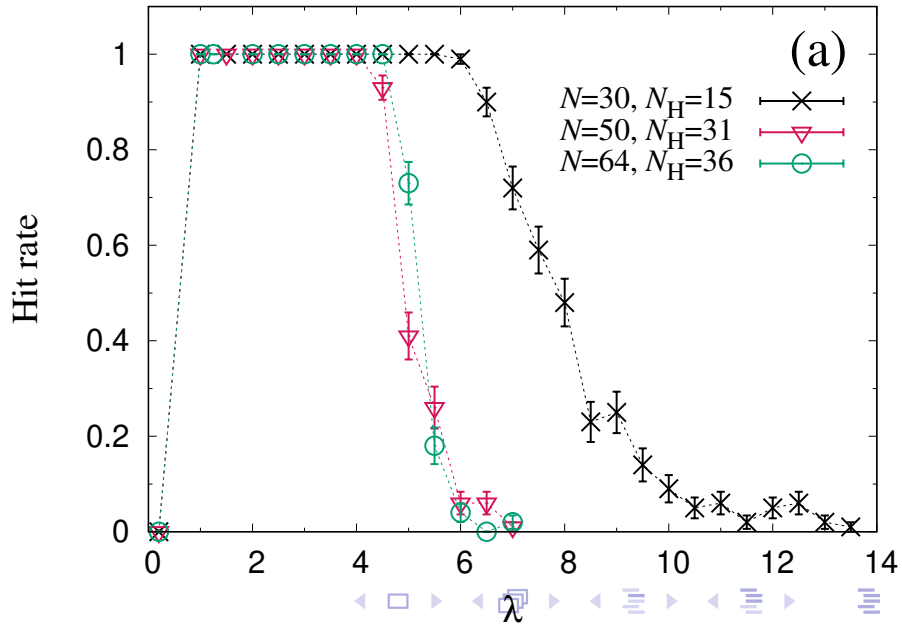


Design: Sequence Optimization with Hybrid

Target structures with $N = 30, 50, 64$.
a few different N_{H} values for each structure
100% success rate in all instances

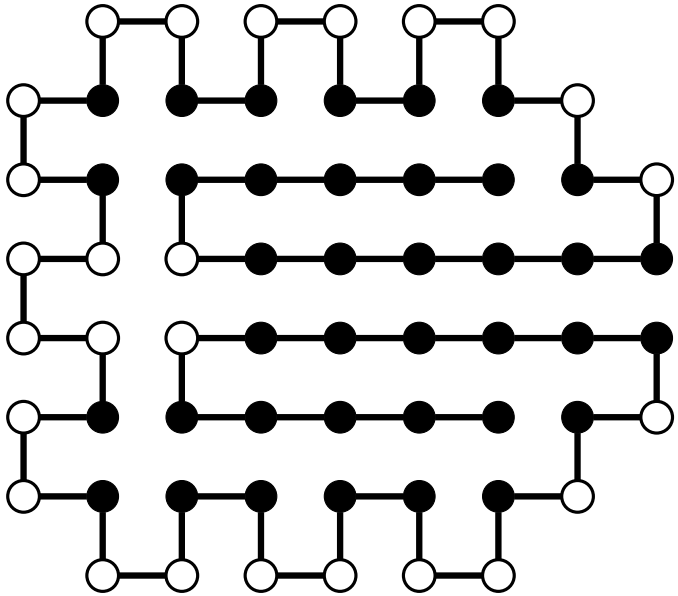


$$E = \sum_{ij} w_{ij} s_i s_j + \lambda \left(\sum_i s_i - N_{\mathrm{H}} \right)^2$$



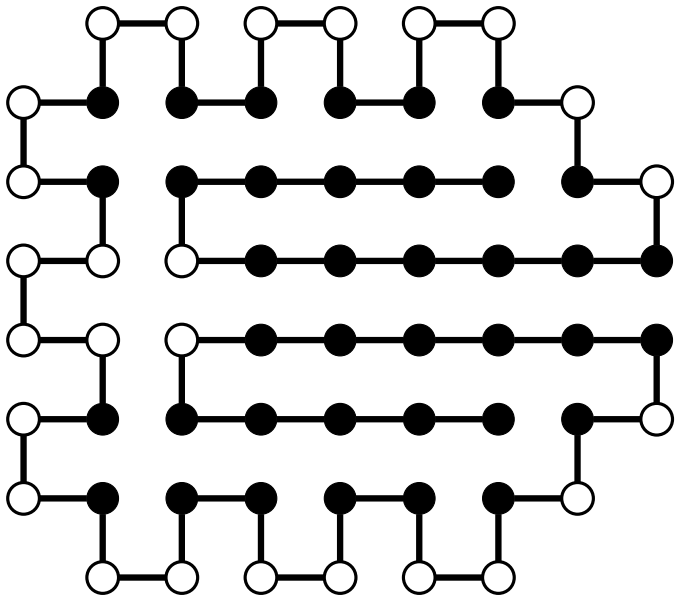
Design: Folding with Hybrid

For every target, at least one sequence found with the intended structure as its unique ground state.



Design: Folding with Hybrid

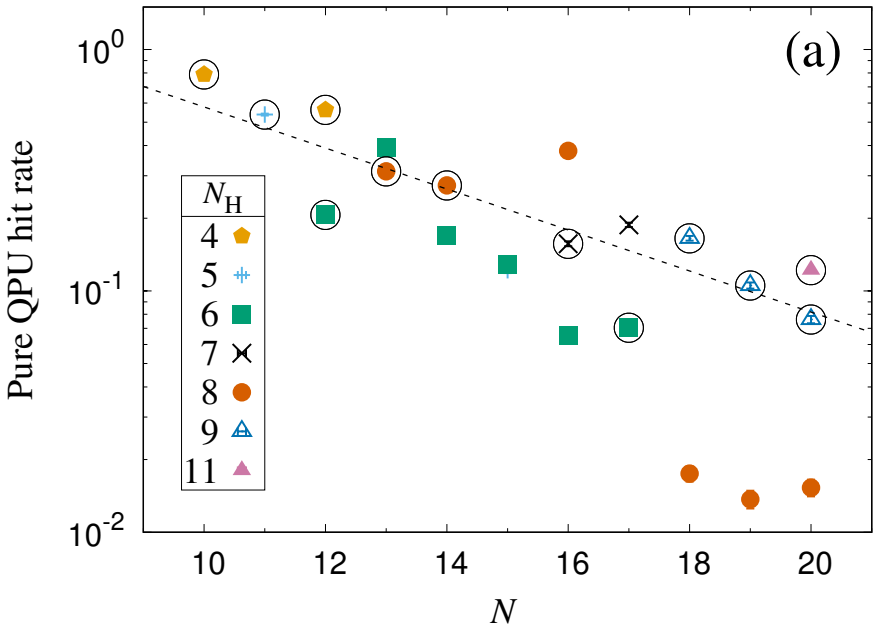
For every target, at least one sequence found with the intended structure as its unique ground state.



Previously studied sequences with $N = 64$ and $N_{\mathrm{H}} = 42$, the ground state is degenerate.

Design: Sequence Optimization with the QPU

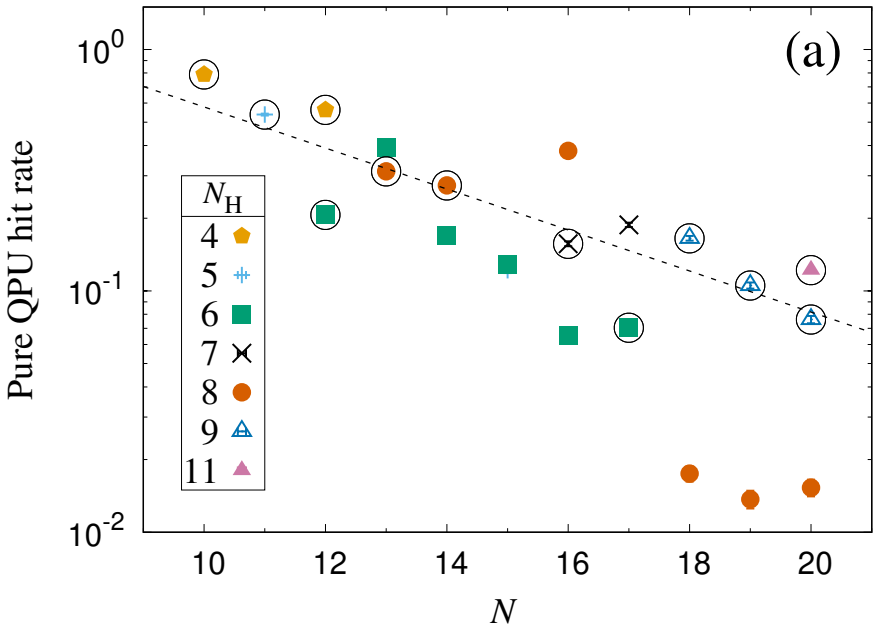
The success rate decays rapidly



Design: Sequence Optimization with the QPU

The success rate decays rapidly

Potential error sources include
(i) thermal noise, (ii) chain breaks, (iii) finite annealing time, (iv) control errors

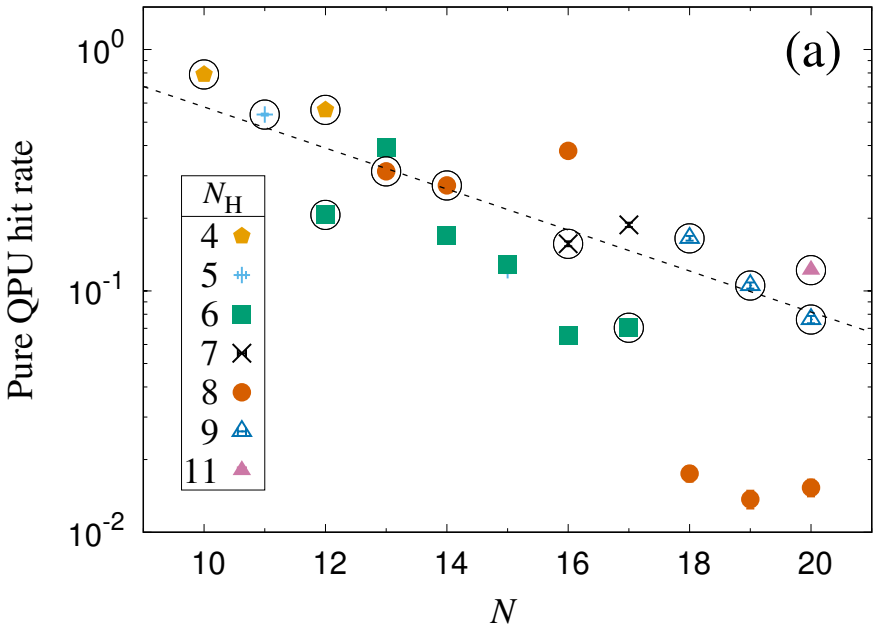


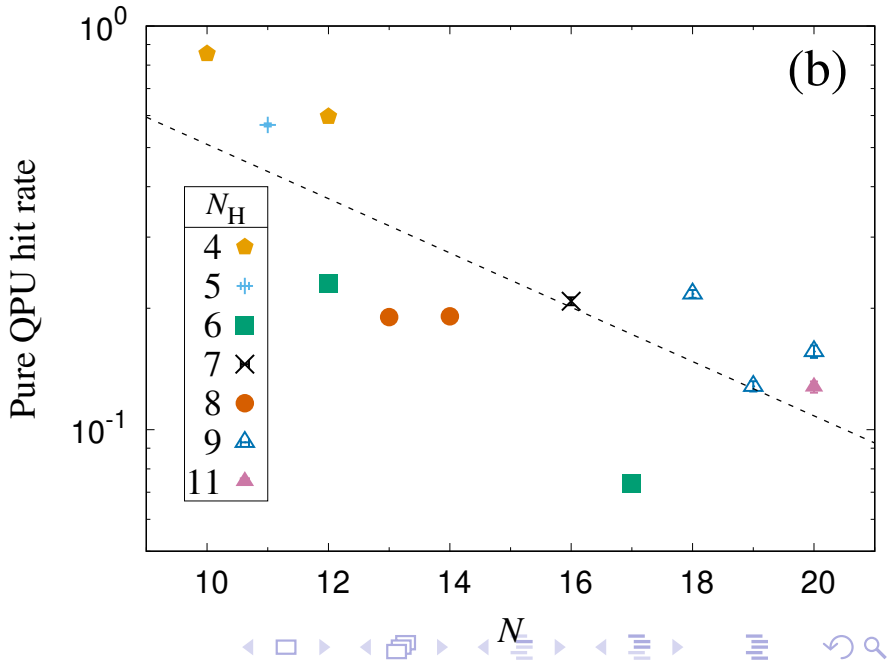
Design: Sequence Optimization with the QPU

The success rate decays rapidly

Potential error sources include
(i) thermal noise, (ii) chain breaks, (iii) finite annealing time, (iv) control errors

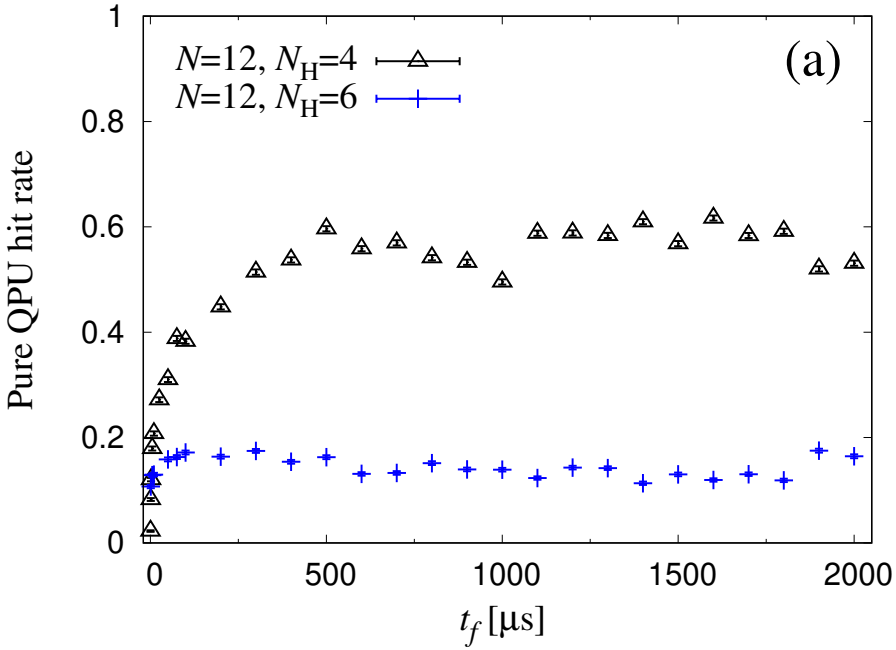
Right shows the data without chain breaks and large energy gaps

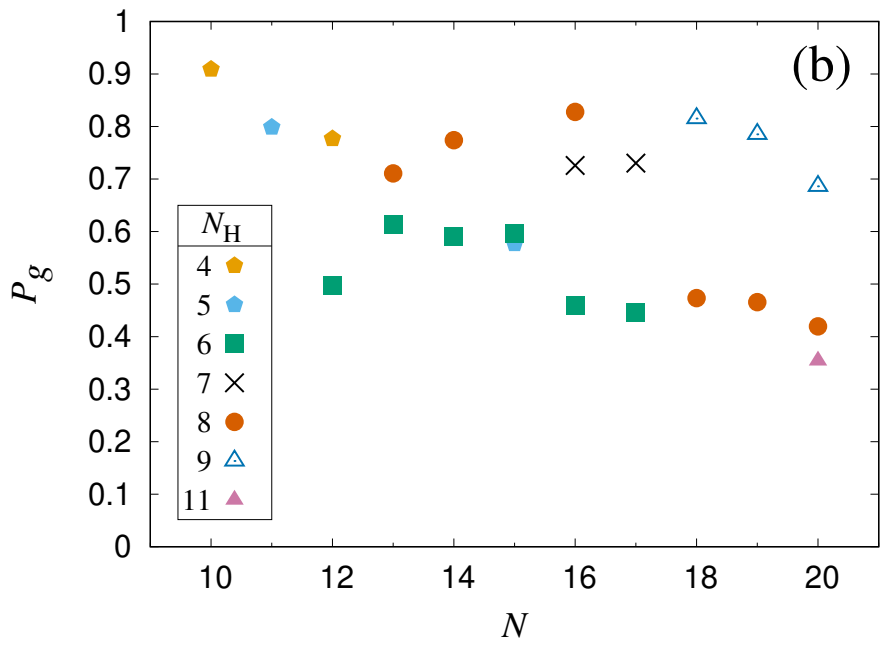




Design: Sequence Optimization with the QPU - finite time

(a) Pure QPU hit rate against annealing time
(b) Success rates when integrating the time-dependent Schrödinger equation for different systems using a fixed annealing time

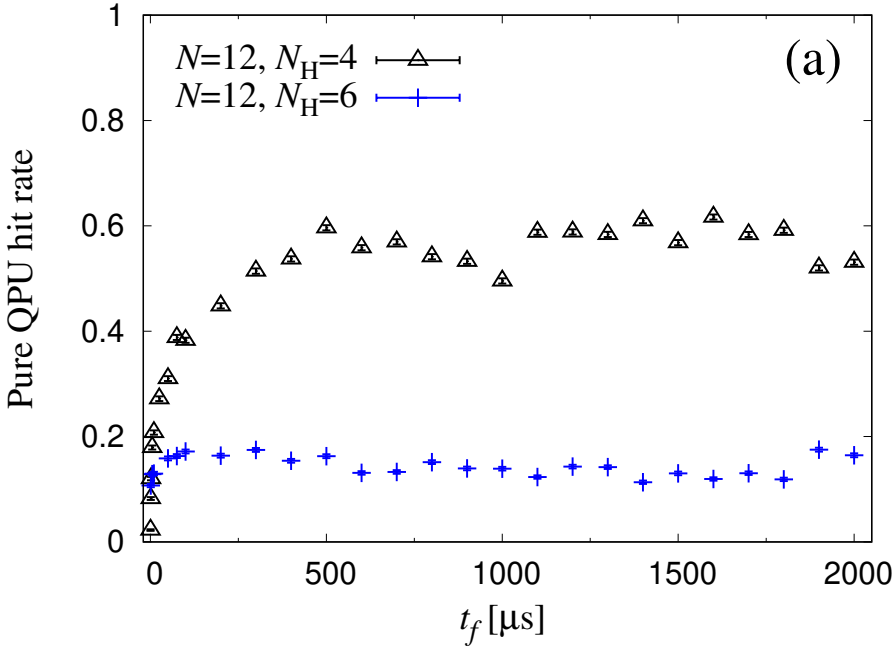


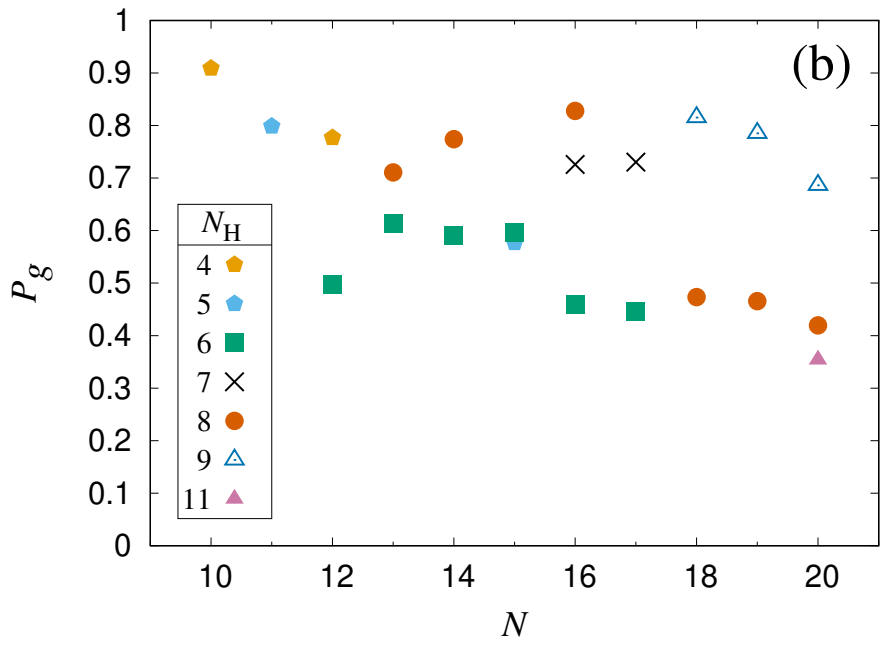


Design: Sequence Optimization with the QPU - finite time

(a) Pure QPU hit rate against annealing time
(b) Success rates when integrating the time-dependent Schrödinger equation for different systems using a fixed annealing time

No indication that finite annealing time explains the decay in success rate





Intermezzo: How does the D-Wave work?

D-Wave uses super-conducting flux qubits with a limited energy range

Intermezzo: How does the D-Wave work?

D-Wave uses super-conducting flux qubits with a limited energy range

Therefore, if you give a problem, $\sum_i h_i \sigma_z^i + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$ with h_is and/or J_{ij}s outside this range, we rescale

Intermezzo: How does the D-Wave work?

D-Wave uses super-conducting flux qubits with a limited energy range

Therefore, if you give a problem, $\sum_i h_i \sigma_z^i + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$ with h_is and/or J_{ij}s outside this range, we rescale

Remember the chains? Several strongly coupled qubits represent one logical qubit.

Intermezzo: How does the D-Wave work?

D-Wave uses super-conducting flux qubits with a limited energy range

Therefore, if you give a problem, $\sum_i h_i \sigma_z^i + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$ with h_is and/or J_{ij}s outside this range, we rescale

Remember the chains? Several strongly coupled qubits represent one logical qubit. The chain coupling strength is often the largest J_{ij} in the reformed problem

Design: Sequence Optimization with the QPU - control errors

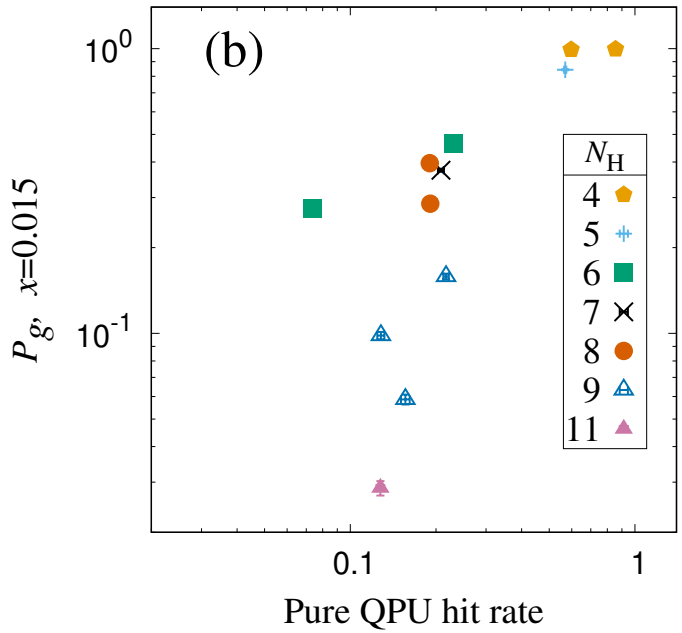
Imperfect implementation if the problem Hamiltonian
$\tilde{H}_P = \sum_i (h_i + \delta h_i)\sigma_z^i + \sum_{ij}(J_{ij} + \delta J_{ij})\sigma_i^z \sigma_j^z$

Assume δh_i, δJ_{ij} independent and Gaussian with std devs σ_h and σ_J

Design: Sequence Optimization with the QPU - control errors

Imperfect implementation if the problem Hamiltonian
$\tilde{H}_P = \sum_i (h_i + \delta h_i)\sigma_z^i + \sum_{ij}(J_{ij} + \delta J_{ij})\sigma_i^z \sigma_j^z$

Assume δh_i, δJ_{ij} independent and Gaussian with std devs σ_h and σ_J
D-Wave: $\sigma_h = x\max|h_i|$ and $\sigma_J = x\max|J_{ij}|$, $x = 0.015$



Semi-quantitative agreement

Design: Conclusion

The hybrid quantum-classical method swiftly and consistently solves the lattice protein folding and design problems for system sizes that are non-trivial with classical methods.

Design: Conclusion

The hybrid quantum-classical method swiftly and consistently solves the lattice protein folding and design problems for system sizes that are non-trivial with classical methods.

The pure QPU results are less impressive. Control errors may have a significant impact on the success rate

Acknowledgements

Acknowledgements: All QA computations were done on the D-Wave Advantage System at the Jülich Supercomputing Centre, Germany.

QAOA: Gate-based optimization

QA: $|f\rangle = \mathcal{T} \exp(\int_0^T H(t)dt) \, |start\rangle$,
QAOA: use $\mathcal{T} \exp(\int_0^T H(t)dt) \approx$
$\exp(-i\gamma_k H_P) \exp(-i\beta_k H_D) \cdots \exp(-i\gamma_1 H_P) \exp(-i\beta_1 H_D)$ when $k \to \infty$

QAOA: Gate-based optimization

QA: $|f\rangle = \mathcal{T}\exp(\int_0^T H(t)dt)\,|start\rangle$,
QAOA: use $\mathcal{T}\exp(\int_0^T H(t)dt) \approx$
$\exp(-i\gamma_k H_P)\exp(-i\beta_k H_D)\cdots\exp(-i\gamma_1 H_P)\exp(-i\beta_1 H_D)$ when $k \to \infty$

run
$|f\rangle = exp(-i\gamma_k H_P)\exp(-i\beta_k H_D)\cdots\exp(-i\gamma_1 H_P)\exp(-i\beta_1 H_D)\,|start\rangle$
and optimize $\vec{\gamma}$ and $\vec{\beta}$ classically

QAOA: Gate-based optimization

QA: $|f\rangle = \mathcal{T}\exp(\int_0^T H(t)dt)\,|start\rangle$,
QAOA: use $\mathcal{T}\exp(\int_0^T H(t)dt) \approx$
$\exp(-i\gamma_k H_P)\exp(-i\beta_k H_D)\cdots\exp(-i\gamma_1 H_P)\exp(-i\beta_1 H_D)$ when $k \to \infty$

run
$|f\rangle = exp(-i\gamma_k H_P)\exp(-i\beta_k H_D)\cdots\exp(-i\gamma_1 H_P)\exp(-i\beta_1 H_D)\,|start\rangle$
and optimize $\vec{\gamma}$ and $\vec{\beta}$ classically

Free to choose $|start\rangle$ and H_D (which causes transitions)

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

Each term in the above Hamiltonian gets its own gate ($\exp(-i\alpha_i H_P)$). That is a lot of gates.

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

Each term in the above Hamiltonian gets its own gate ($\exp(-i\alpha_i H_P)$). That is a lot of gates.

Make a graph with spins as nodes and the quadratic terms in E_is as edges

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

Each term in the above Hamiltonian gets its own gate ($\exp(-i\alpha_i H_P)$). That is a lot of gates.

Make a graph with spins as nodes and the quadratic terms in E_is as edges

Remember that H_D causes transition. We choose an H_D that only move between sets of spins with no edges in common.

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

We choose an H_D that only move between sets of spins with no edges in common.

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

We choose an H_D that only move between sets of spins with no edges in common.

There is such a Hamiltonian called the MIS-mixer

QAOA: Folding [Manuscript in preparation]

In collaboration with Leif Gellersen and Stefan Prestel

$$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s' \rangle} \sigma_s^f \sigma_{s'}^{f'}$$

$$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2, \qquad E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2}$$

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1}$$

We choose an H_D that only move between sets of spins with no edges in common.

There is such a Hamiltonian called the MIS-mixer

If we combine this with divide-and-conquer methods we can do large proteins with few qubits and gates