LU TP 22-35
June 2022

Folding HP Lattice Proteins on a Quantum Annealer

Lucas Knuthson

Department of Astronomy and Theoretical Physics, Lund University

Master thesis supervised by Anders Irbäck

Abstract

Quantum annealing provides a promising avenue for obtaining good approximate solutions to difficult optimization problems. Protein folding represents one such problem. For testing novel algorithms and technologies, simplified lattice models are well suited, as they represent considerable computational challenges despite their simplicity. One such model is the HP model, where the protein is represented as a self-avoiding chain of hydrophobic (H) and polar (P) beads residing on a lattice. Previous attempts at folding lattice proteins on a quantum annealer used chain growth techniques, where self-avoidance is tricky to implement. In this project, we develop a novel spin representation of the HP model suited for quantum annealing. This approach naturally handles self-avoidance, and performs well in terms of scaling properties with chain length. The approach is implemented using classical simulated annealing, a hybrid quantum-classical approach and pure quantum annealing. In the pure quantum annealing case, we successfully fold the largest chain done on a quantum computer. However, we also notice that pure quantum annealing can not match simulated annealing yet. In contrast, the hybrid approach is able to solve the largest HP chains, $N = 30$, where N is the number of beads, for which exact solutions are known. Moreover, it outperforms classical simulated annealing using the same encoding, both in terms of success rate and solution time, and successfully compare with the best-known solutions for larger chains, $N = 48$ and $N = 64$. Further, we see that the encoding is robust in terms of changes in the parameters required to constrain the spin system to chain-like configurations. The calculations were performed on a D-Wave Advantage quantum annealer.

List of acronyms

SA - Simulated Annealing
QUBO - Quadratic Unconstrained Binary Optimization
QPU - Quantum Processing Unit

Populärvetenskaplig beskrivning

Kvantdatorer har fått mycket fokus i media de senaste åren på grund av sin potentiella förmåga att lösa problem som är för tidskrävande för att låta sig lösas på klassiska datorer. Kvantdatorer skiljer sig i grunden från dagens datorer, genom att bygga på gåtfulla kvantmekaniska egenskaper. En sådan är kvantsuperposition, som innebär att en partikel kan vara i flera kvanttillstånd samtidigt. Antag, till exempel, att uppgiften är att finna vägen ut ur en labyrint. En klassisk dator löser detta problem genom att testa alla möjliga vägar en efter en. En kvantdator skulle, genom kvantsuperposition, istället kunna testa alla vägar samtidigt, och därmed lösa problemet snabbare.

Att bygga en kvantdator är en utmaning eftersom processorn måste vara isolerad från världen runtomkring för att behålla sina kvantmekaniska egenskaper. De stora satsningarna som för närvarande görs fokuserar därför inte bara på generella kvantdatorer, utan även på mer specialicerade sådana för särskilda ändamål. Ett viktigt exempel är adiabatiska kvantdatorer för optimeringsproblem. En sådan dator består av ett kvantsystem där energifunktionen av systemet kan regleras. Initialt kan kvantsystemet antas befinna i ett tillstånd som i labyrintexemplet motsvarar en blandning av alla möjliga vägar. Det går att visa att tillståndet sedan kan ändras till ett som motsvarar vägen ut ur labyrinten, genom att ändra energifunktionen. Om denna ändring sker långsamt nog hittar vi vägen ut varje gång. Dessvärre kan förändringen i praktiken inte ske hur långsamt som helst, eftersom processorn måste hållas helt isolerad. Detta medför att den sökta vägen ut inte kommer hittas med 100% sannolikhet.

I detta projekt försöker vi använda en adiabatisk kvantdator för att bestämma strukturen av förenklade proteiner. Proteinstrukturberäkningar är ett svårt men viktigt problem, eftersom kännedom om strukturen är avgörande för att förutspå hur exempelvis enzymproteiner fungerar. I vår förenklade modell kan proteinet ses som ett kapat pärlhalsband, där pärlorna representerar aminosyror. Dessutom tillåter vi bara pärlorna att vara på specifika platser, precis som schackpjäser bara får vara på en utav 64 rutor på schackbrädet. Trots denna restriktion kan pärlhalsbandet anta en mängd olika former. Bland dessa söker vi den struktur som har lägst energi. Försök att lösa detta optimeringsproblem på en kvantdator har gjorts förut. I denna uppsats utvecklar vi en ny formulering av strukturbestämmningsproblemet, väl lämpad för kvantdatorer. Vi testar med framgång metoden på en kvantdator utvecklad av D-Wave Systems, för adiabatiska kvantberäkningar.

Contents

1 Introduction

Quantum computers provide a promising avenue for solving difficult optimization problems since they can utilize quantum phenomena such as superposition, entanglement and quantum tunneling. These phenomena imply that NP problems can, potentially, be solved in polynomial time [1,2]. The development of these computers has taken two major tracks, quantum annealers [3] and gate-based systems [4]. In this project, we have used a quantum annealer manufactured by D-Wave (D-Wave Systems Inc.). The D-Wave machines are made up of a spin system and minimize energy functions in the form of Ising spin glass Hamiltonians. Therefore, problems have to be mapped onto this form. Mapping difficult optimization problems onto Ising spin glass system has seen success [5–8], and goes back to the eighties [7,8].

Protein folding, going from sequence to structure by minimizing an energy function, represents a difficult optimization problem. Simplified lattice-based models can often provide qualitatively relevant results. However, these simplified models still represent a computational challenge and are therefore ideal testbeds for new algorithms. A pioneering formulation of the lattice protein folding problem in terms of an Ising spin glass Hamiltonian was given by Perdomo and coworkers [9] as a quadratic unconstrained binary optimization (QUBO) problem. They used the HP model [10], where a protein is represented by a chain of N beads, either being hydrophobic (H) or polar (P), residing on a lattice. This formulation required a significant amount of qubits for small chains, yet scaled polynomially.

An early attempt to fold a short lattice protein ($N = 6$) using quantum annealing was carried out on a D-Wave machine [11]. The implementation used a growth algorithm, where the turns along the chain were mapped onto qubits. This implementation was resource-efficient for small chains but scaled exponentially with N. Recently, a similar encoding was implemented on IBM's gate-based quantum computers for a short chain ($N = 7$) [12]. These algorithms offer a compact chain representation, but make self-avoidance difficult to implement. For a recent review of lattice protein folding on quantum computers, see ref. [13].

In contrast to refs. [11, 12], in this project, we propose a different mapping for lattice proteins. In our mapping all sites on the lattice hosts qubits. The approach was inspired by ref. [14], which used a D-Wave machine to sample homopolymers and shares similarites with a QUBO formulation for lattice heteropolymers [15] which, to our knowledge, was never implemented. Our approach is such that minimizing the energy function, E, makes the qubits coalesce to a finite set active qubits defining the desired folded structure. Further, the function E includes a self-avoidance term and, importantly for quantum annealing, is quadratic (or 2-local) in the binary variables. This dynamical encoding can be seen as a clustering approach driven by requiring a legal chain on the lattice.

We evaluate the performance of the proposed method using the HP model [10] as a testbed. We do this by using classical simulated annealing (SA) [16], followed by explorations on the D-Wave Advantage annealer [17]. To measure the performance of the method we compare with synthetic HP [10] sequences with up to 30 beads, where the exact solution is known [18,19] from exhaustive enumeration. In addition, we also present results obtained for two longer sequences [20] with 48 and 64 beads, respectively, which have been studied by various classical methods.

This thesis is organized as follows. Sec. 2 presents a short introduction to quantum computation basics, while in sec. 3 we discuss the D-Wave annealers. In sec. 4 the HP model is presented along with the novel energy function, E, and sec. 5 presents the computational methods used in this project. The results and evaluation of the model are presented in sec. 6. The thesis ends with a few concluding remarks and an outlook, given in sec. 7.

2 Quantum Computing Basics

Quantum computing has received much attention recently, and progress is being made. For a more detailed discussion on the topic, the reader is referred to [1,2,21–25]. This section outlines some basics of the field. In this work we set $\hbar \equiv 1$. In quantum computing, to process information, the quantum bit (qubit) is used as the unit for computation [2]. Contrary to the classical binary so-called bit, the qubit can utilize quantum phenomena such as superposition and entanglement. It can be seen as a generalization of the bit. The qubit $|\psi\rangle$ is often described using two complex numbers ψ_0, $\psi_1 \in \mathbb{C}$, such that [1,2,21]

$$|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} \tag{2.1}$$

where the state is normalized such that $\langle\psi|\psi\rangle = |\psi_0|^2 + |\psi_1|^2 = 1$. The last equality in eq. (2.1) is only true in the $|0\rangle, |1\rangle$ basis, where these states can be represented by the standard basis vectors [2,21]

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.2}$$

Actual calculations use multi-qubit systems.

Similarly to the one qubit case, a multi-qubit system can also be described using a set of complex coefficients. However, for a system of n qubits, 2^n coefficients are needed instead of two. Thus, the system $|\psi\rangle$ can be described using ψ_0,...,ψ_{2^n-1} as [2,21]

$$|\psi\rangle = \psi_0 |0\cdots00\rangle + \psi_1 |0\cdots01\rangle + \ldots + \psi_{2^n-1} |1\cdots11\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{2^n-1} \end{pmatrix}. \tag{2.3}$$

The basis states in eq. (2.3) are Kronecker products of the single qubit basis states seen in eq. (2.2). For example, the $|q_0 q_1 \cdots q_{n-1}\rangle = |q_0\rangle \otimes |q_1\rangle \otimes \ldots \otimes |q_{n-1}\rangle$, where $q_i = 0, 1$ for the ith qubit of the system. Note that the basis state $|q_0 q_1 \cdots q_{n-1}\rangle$ with the coefficient ψ_j contains the binary value of j inside the ket, i.e. $\text{bin}(j) = q_0 q_1 \cdots q_{n-1}$. Therefore, the state $|\psi\rangle$ can be written as [2, 21]

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \psi_j |j\rangle \tag{2.4}$$

In recent years two major quantum computing paradigms have emerged. These are *gate based quantum computing* [4] and *quantum annealing* [3] (also known as *adiabatic quantum computing*)[1]. Gate based quantum computing works by applying operators, also called gates, onto the state ket of the system such that the final ket is the desired state [1,21]. This paradigm has received much attention and is the focus of Google's, Microsoft's and IBM's computers. Since such a computer was not used in this thesis, it will not be discussed any further. The second paradigm uses the adiabatic theorem, which is discussed in sec. 2.1. The somewhat simpler technology of the quantum annealers implies that they are less difficult to manufacture, and as such, quantum annealers with 5000+ qubits are available already [17]. Thus, quantum annealers are approaching sizes sufficiently large to tackle real-world problems. It can also be shown that both paradigms, theoretically, have the same computational power [22].

2.1 Quantum Annealing

The idea of quantum annealing machines is to use the adiabatic theorem in quantum mechanics. The theorem assumes a Hamiltonian of the form [1, 21, 22]

$$H(t) = A(t)H_0 + B(t)H_1, \quad t \in [0, \tau]. \tag{2.5}$$

The coefficient in front of H_1 goes from zero to unity, while the coefficient in front of H_0 goes from unity to zero. Therefore, $H(0) = H_0$ and $H(\tau) = H_1$. The theorem states that if the system is in the non-degenerate ground state of $H(t)$ at $t = 0$, and $H(t)$ is changing slowly, the system will remain in the instantaneous ground state a time τ later. For the theorem to work the anneal time has to be chosen such that [22]

$$\tau \gg \max_{0 \leq s \leq 1} \frac{|\langle\psi_1(s)| \frac{dH(s)}{ds} |\psi_0(s)\rangle|}{(E_0(s) - E_1(s))^2}, \quad s \equiv \frac{t}{\tau} \tag{2.6}$$

where $|\psi_0(s)\rangle$ ($|\psi_1(s)\rangle$) and $E_0(s)$ ($E_1(s)$) are the instantaneous ground (first excited) state and its energy, respectively. Eq. 2.6 implies that H_0 and H_1 in eq. 2.5 have to be chosen

[1] There are other paradigms as well, such as measurement based quantum computing [26] or fusion-based quantum computing [27]

such that they do not commute. If they commute there will be a time, s_1, where the energy gap $E_0(s_1) - E_1(s_1)$ vanishes [1].

To solve problems using the adiabatic theorem the state Hamiltonian H_0 is chosen such that the ground state is easy to construct, while H_1 is chosen such as to have the desired solution as its ground state. On the D-Wave machines, the Hamiltonian H_0 is chosen to be [1, 21]

$$H_0 = \sum_{i=0}^{n-1} h_i \sigma_i^x, \tag{2.7}$$

for a system of n qubits, where $h_i < 0$ for all i. Here $\sigma_i^x = I \otimes ... \otimes \sigma^x \otimes ... \otimes I$, with I being the identity matrix and σ^x being the x Pauli matrix placed at position i in the product. This implies that the system starts in the state [1]

$$|\psi(0)\rangle = \left(\frac{1}{\sqrt{2}}\right)^n (|0_0\rangle + |1_0\rangle) \otimes \cdots \otimes (|0_{n-1}\rangle + |1_{n-1}\rangle) \tag{2.8}$$

in the σ^z basis, where the $|0\rangle$ ($|1\rangle$) state is interpreted as the spin-↑ (spin-↓) state from quantum mechanics. Therefore, the system starts out in an equal superposition state in the computational basis. Moreover, on the D-Wave machines the Hamiltonian H_1 is chosen to be an Ising spin glass Hamiltonian [1]

$$H_1 = \sum_{i=0}^{n-1} h_i \sigma_i^z + \sum_{i<j}^{n-1} J_{ij} \sigma_i^z \sigma_j^z, \tag{2.9}$$

such that its ground state, $|q_0 ... q_{n-1}\rangle$, encodes the solution to the proposed problem. Since the system starts in the superpostition state of eq. 2.8, it can use quantum tunneling to end up in the ground state of the Hamiltonian in eq. 2.9 Therefore, if an optimization problem can be recast into an Ising spin glass Hamiltonian, with the solution being the ground state, it can theoretically be solved by a quantum annealer.

3 The D-Wave Machines

D-Wave Systems is a Canadian company that makes commercially available heuristic solvers that use quantum annealing. Here, we will briefly describe how they work, for a more detailed description of the physics of the system, the reader is referred to refs. [23–25]. The qubits in these machines are called flux qubits, and are built up of superconducting circuits. The flux qubits are connected by so-called couplers, and both the couplers and flux qubits are controlled by external magnetic fields. The parameters h_i and J_{ij} of the final state Hamiltonian in eq. 2.9 are controlled by time-independent magnetic fields for the qubits and couplers, respectively. Further, the annealing process is controlled by a time-dependent magnetic fields which change the Hamiltonian in eq. 2.7 to the one in eq. 2.9 [21]. The collection of qubits and couplers is called the quantum processing unit (QPU).

The QPU is subject to several limiting factors that may influence the result of the annealing process. These include factors originating from the environment, such as the temperature which affects the superconductivity of the qubits, or high energy photons that may cause unwanted excitations of the qubits [28]. Other factors, that do not originate from the environment, are the so-called integrated control errors [29]. These errors come from an imperfect implementation of eq. 2.9, giving the new Hamiltonian [29, 30]

$$H_\delta = \sum_{i=0}^{n-1}(h_i + \delta h_i)\sigma_i^z + \sum_{i<j}^{n-1}(J_{ij} + \delta J_{ij})\sigma_i^z\sigma_j^z, \tag{3.10}$$

where δh_i and δJ_{ij} are noise terms. When this occurs the D-Wave machine finds the ground state of eq. 3.10 instead of eq. 2.9 [30]. It has been shown that the probability of eq. 3.10 and eq. 2.9 sharing the same ground state shrinks exponentially with increasing n [30]. Therefore, this causes an exponential decrease in the amount of successful anneals when increasing the system size.

Besides the aforementioned limitations, the D-Wave also faces challenges due to the sparse connectivity of the QPU. The most recent QPU is called Advantage and it contains 5000+ qubits and up to 15 couplers per qubit. The topology of Advantage is a Pegasus graph of size 16 (P_{16}) [17, 21] (see Fig. 1a for a P_4 graph). The sparse connectivity of the Pegasus graph implies that not all problems can be mapped onto a Pegasus subgraph. To circumvent this issue D-Wave embeds the problem graph in the QPU by forming "chains" of qubits. These chains consist of several qubits that have a strong coupling between them, such that they effectively act as a single qubit (see Fig. 2). It is known that the upper bound of extra qubits needed scales quadratically with system size [31]. The effective qubits corresponding to chains can have a higher connectivity than that of the original qubits, meaning that more problems can be solved at the cost of using more qubits [32].

3.1 Programming on the D-Wave Quantum Annealers

In this section we briefly describe how to program on the D-Wave machines using the Python package `Ocean-SDK` [33, 34], supplied by D-Wave. For more detail, the reader is referred to ref. [21]. To minimize a function using the QPU, it has to be submitted as a QUBO model [21]

$$\sum_{i=0}^{n-1} a_i x_i + \sum_{i<j}^{n-1} b_{ij} x_i x_j = \sum_{i\leq j}^{n-1} x_i Q_{ij} x_j, \quad x_k = 0, 1 \tag{3.11}$$

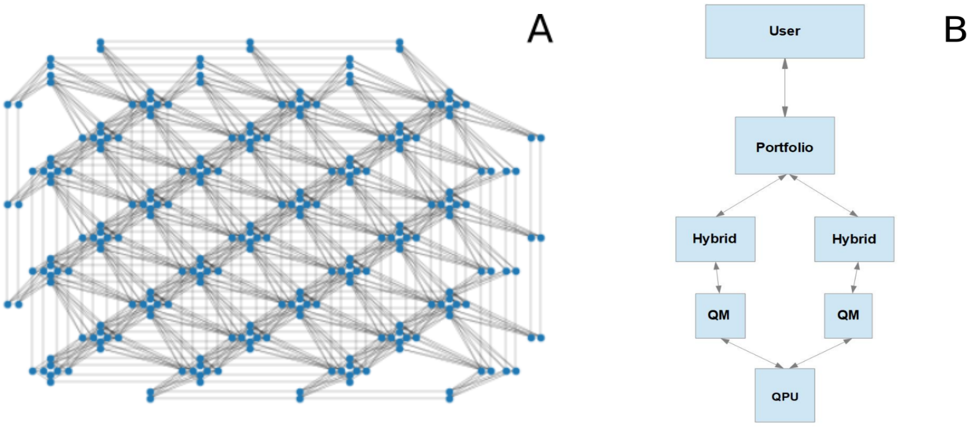

Figure 1: (A) Pegasus graph of size 4 (P_4) where the nodes are shown as blue dots and the edges as gray lines. (B) A schematic view of the hybrids solvers. The portfolio solver reads the input from the user, starts several hybrid solvers and calculates a time limit. The hybrid solvers contain a classical algorithm and a quantum module (QM). The classical solvers attempt to solve the problem, while the QM partitions the problem into subproblems and sends them to the QPU. The QPU responds with the solutions to the subproblems, and these responses helps to guide the classical solver. At the time limit, the hybrid solvers deliver their results to the portfolio, which picks the best solution and send it to the user.

where the equality is true since $x_k = 0, 1$ which implies that $x_k = x_k^2$, or as an Ising Hamiltoninan [21]

$$\sum_{i=0}^{n-1} h_i s_i + \sum_{i<j}^{n-1} J_{ij} s_i s_j, \quad s_k = -1, 1 \tag{3.12}$$

where the solution is the vector $\vec{x}$ ($\vec{s}$) that minimizes eq. 3.11 (eq. 3.12). The functions are communicated to the QPU as a dictionary containing the coefficients. The dictionary either takes the form `Q`$= \{(i, j) : Q_{ij}\}$ for QUBO or as two dictionaries `h`$=\{i : h_i\}$ and `j`$= \{(i, j) : J_{ij}\}$ for Ising. Further, the class `DWaveSampler` [33–35] chooses which QPU to send the problem to (in our case it is the Advantage). This sampler has two functions, `sample_qubo` and `sample_ising`, which submit the dictionary `Q`, or `h` and `j` to the chosen QPU [21]. These functions also submit additional parameters such as the `Annealing_time`, τ, specified in μs, and the number of samples obtained per submission (`num_reads`), which must be $< 10^6/(\tau/\mu\text{s})$. Typically, these parameters have to be optimized for each individual problem.

As stated previously, if the problem has higher connectivity than the QPU it has to be embedded in the QPU. Generating an optimal embedding is itself NP-hard, thus `Ocean` offers a heuristic algorithm to generate a Pegasus embedding. This algorithm uses the class

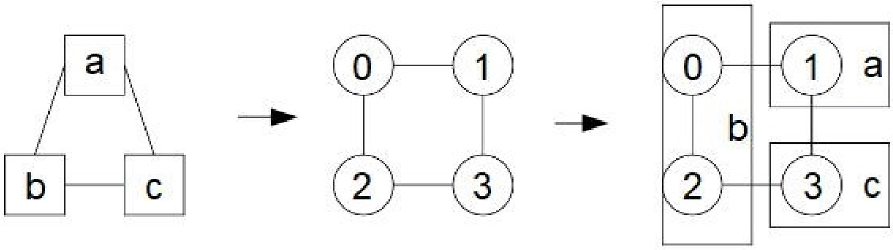

Figure 2: Schematic view of how an embedding is made on the D-Wave machines. A triangular graph with nodes a,b and c is embedded in a square graph, with nodes 0,1,2 and 3, by letting 0 and 2 ”play” the role of b. In this case, 0 and 2 form a chain of length 2. The triangular graph is the problem graph and the graph furthest to the right represents its embedding into the QPU topology.

`EmbeddingComposite` [33, 34], which allows for probabilistic generation of embeddings or a user-submitted one. The strength of the biases for the chains in the embedding is regulated by the parameter `chain_strength`, which is often chosen to be the smallest value that does not cause chains to break. This parameter is submitted, along with the other parameters, using the functions `sample_qubo` and `sample_ising`. `Ocean` also offers classes that combine `DWaveSampler` with a specific embedding. One such class is `DWaveCliqueSampler`, which combines `DWaveSampler` with an embedding for a fully connected graph. This class is designed for dense problems [35].

3.2 Hybrid Quantum-Classical Computations

Besides solvers based entirely on quantum annealing, D-Wave also offers hybrid quantum-classical solvers. The hybrid approach uses classical solvers while sending subproblems of the submitted problem as queries to the QPU. The subproblems are constructed using classical heuristic partitioning algorithms and are small enough to be solved effectively on the QPU. The solutions to the subproblems serve to guide the classical solvers [36] (see Fig. 1b). The goal is to speed up the solution of challenging problems by queries to the QPU. With the hybrid approach, it is possible to tackle much larger problems, with many thousands of fully connected variables, than what can be dealt with using quantum annealing alone. However, the internal function of the hybrid solvers is proprietary information (see the documentation on Leap Hybrid solvers [35, 36]).

4 Lattice Protein Folding and its QUBO Encoding

We intend to tackle the problem of folding HP lattice proteins [10] using quantum computing. Therefore, in this section, we present it along with our novel QUBO encoding of the HP model, which we used to obtain the results presented in sec. 6.

4.1 The HP Model

The HP model is a lattice protein model [10], where the protein is represented as a self-avoiding chain of hydrophobic (H) and polar (P) beads residing on a lattice [10, 18, 19]. The energy function of the protein is computed as a sum of interaction energies between all beads. Therefore, the energy function takes the form [19]

$$E = \sum_{f<f'} U(h_f, h_{f'}) C_{ff'} \tag{4.13}$$

where h_f denotes the type of bead (H or P) at position f in the sequence and $C_{ff'}$ is defined as [19]

$$C_{ff'} = \begin{cases} 1, & \text{if beads } f, f' \text{ are nearest neighbors on the lattice but } \; |f - f'| > 1. \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

The function $U(h_f, h_{f'})$ determines the interaction energy, which is often set to -1 if h_f and $h_{f'}$ are both H, while being 0 otherwise. Therefore, in this convention, the energy of the protein is $-N_{\text{HH}}$, where N_{HH} is the number of HH nearest-neighbor contacts.

In part due to its simplicity, the HP model has been extensively studied [18–20, 37]. In particular, through exhaustive enumeration, all HP sequences which have a unique minimum-energy structure, and this structure, have been determined for chains with $\leq$ 30 beads [18, 19]. This enumeration also showed that several sequences can have the same unique minimum-energy structure. Such a sequence is said to design that structure and the number of sequences that designs a given structure is called the designability of the structure. Structures with high designability thus show robustness to mutation [19]. Further, on a 2D square lattice approximately 2% of all HP sequences with $\leq$ 30 beads have well-defined minimum-energy structures [18]. These ground state structures, and other low energy configurations, tend to display a hydrophobic core of H beads.

Besides the exact enumerations, Monte Carlo methods have been employed to find low-energy structures. In particular, sequences of lengths 48 and 64 beads have been studied [20, 37, 38]. However, the structures found using these methods are not guaranteed to be the ground state structure.

4.2 Binary Quadratic Model for HP Lattice Proteins – QUBO Encoding

Given an HP sequence $(h_1 \ldots, h_N)$, $h_i \in \{\mathrm{P}, \mathrm{H}\}$, we wish to determine its ground state using quantum annealing. To this end, we present a binary encoding for HP lattice proteins, assuming a square grid with L^2 sites.

Inspired by the binary representation of homopolymers of ref. [14], instead of directly encoding chain configurations, we introduce fields of binary variables along with penalty terms, the latter of which serve to ensure that the final binary field configurations correspond to proper chain configurations. To reduce the number of binary variables, we make a checkerboard division of the lattice into even and odd sites, and use the fact that in a valid chain configuration all even (odd) beads share the same lattice site parity (see Fig. 3). Therefore, we may assume that even (odd) beads reside on even (odd) lattices sites. Thus, we introduce one set of binary fields, σ_s^f, to describe the location of even beads, and another set for odd beads $\sigma_{s'}^{f'}$. Here, the indices f and s run over even beads and sites, respectively, while f' and s' run over odd beads and sites. We set $\sigma_s^f = 1$ if bead f is located on site s, and $\sigma_s^f = 0$ otherwise. The odd fields $\sigma_{s'}^{f'}$ are defined in the same way. The division into even and odd sites reduces the number of variables required from $N \times L^2$ to $\approx N \times L^2/2$.

Having defined the degrees of freedom, we now describe the energy function. In our QUBO model, the total energy E has the form

$$E = E_{\mathrm{HP}} + \sum_{i=1}^{3} \lambda_i E_i \,, \tag{4.15}$$

where E_{HP} is the energy of the HP model (see above) and the remaining three terms E_1, E_2 and E_3 are constraint energies. The strengths of the constraints are set by the parameters λ_i.

Specifically, in terms of the binary fields, the four energies can be expressed as follows.

- The HP energy $E_{\mathrm{HP}} = -N_{\mathrm{HH}}$ can be rewritten as

  $$E_{\mathrm{HP}} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s,s'\rangle} \sigma_s^f \sigma_{s'}^{f'}, \tag{4.16}$$

  where the second sum runs over all nearest-neighbor pairs of sites, $\langle s, s'\rangle$. Such a pair always consist of one even and one odd site. The beads f and f' must both be of type H for a non-zero energy contribution, and must not, with our definition of a contact, be adjacent along the chain.

- The first constraint energy, E_1, is given by

  $$E_1 = \sum_f \left(\sum_s \sigma_s^f - 1 \right)^2 + \{\text{same for odd parity}\} \,, \tag{4.17}$$

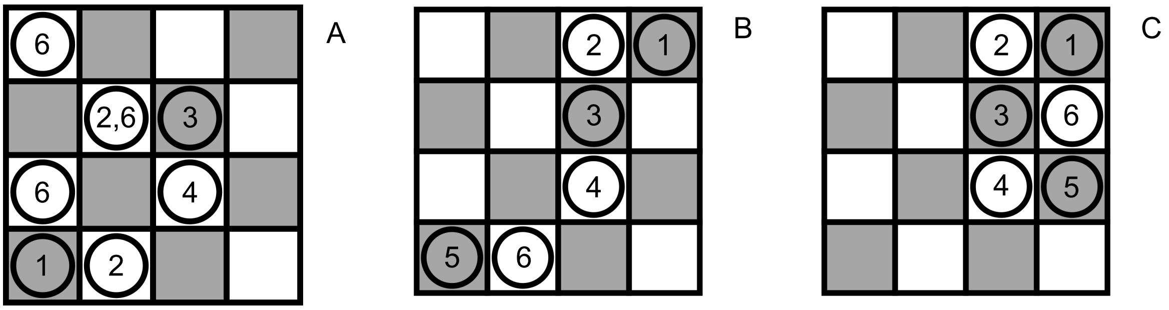

Figure 3: Evolution of the binary model described in Sec. 4.2 in a hypothetical SA run for the 6-bead sequence HPHPPH. Circles represent beads, and numbers indicate bead positions along the sequence. By construction, odd/even beads can reside only on grey/white sites. (A) Early stage. Typically, all the three constraints are violated ($E_1, E_2, E_3 > 0$). (B) Intermediate stage. Some but not all of the constraints are satisfied (in this example: $E_1 = E_2 = 0$, $E_3 > 0$). (C) The final state, in this example corresponding to the desired minimum-energy structure of the given sequence ($E_{\text{HP}} = -2$, $E_1 = E_2 = E_3 = 0$).

and serves to ensure that each bead is located at exactly one lattice site.

- The energy E_2 makes the chain self-avoiding. It is given by

$$E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \sigma_s^{f_1} \sigma_s^{f_2} + \{\text{same for odd parity}\}\,, \tag{4.18}$$

and provides an energy penalty whenever two beads occupy the same site.

- The final energy, E_3, has the form

$$E_3 = \sum_{1 \leq f < N} \sum_s \sigma_s^f \sum_{|s'-s|>1} \sigma_{s'}^{f+1} + \{\text{same with odd/even parity interchanged}\}\,, \tag{4.19}$$

and is responsible for connecting the beads to a chain. It provides an energy penalty whenever two adjacent beads along chain are not nearest neighbors on the lattice.

It is worth noting that we have a physical chain configuration if and only if $E_1 = E_2 = E_3 = 0$.

As indicated in sec. 1, the above binary model shares similarities with the "diamond" encoding proposed in ref. [15]. The latter method is able to reduce the number of binary variables required for very short chains, by fixing the position of the first bead and using the fact that odd and even beads can be assumed to belong to different layers of the "diamond". For long chains, our choice of a freely moving chain on a simple odd/even checkerboard is more resource-efficient since, in general, the search for the ground state can be carried out on a smaller grid if the chain is freely moving. Besides scaling, our constraint energies E_i also differ from those of ref. [15]. In our encoding, all three constraint energies are manifestly non-negative for both physical and unphysical spin configurations, which makes our method more robust to changes in the strength parameters λ_i . Since the encoding in ref. [15] was never implemented, its dependence on the parameters is not known.

5 Computational Methods

5.1 Simulated Annealing

Before turning to quantum annealing, we tested this QUBO model using SA, with the system defined by the partition function $Z = \sum_{\{\sigma_s^f, \sigma_{s'}^{f'}\}} e^{-\beta E}$, where β denotes inverse temperature and E is given by eq. 4.15. All runs spanned the same set of 25 temperatures, given by $\beta_0 = 1$ and $\beta_{i+1} = 1.05\beta_i$. At each temperature, 10^4 sweeps were performed, where one sweep comprises, on average, one attempted update per spin variable. The updates were single-spin flips, controlled by a Metropolis acceptance criterion. All runs were started from random initial spin configurations, and used a 10^2 grid.

For comparison, we also conducted SA runs based on the conventional explicit-chain representation of the HP model. Here, the energy was given by E_{HP}, without the constraint terms. The set of temperatures was the same as in the QUBO SA runs. The simulations used three Metropolis-type elementary moves: local one- and two-bead updates, and a non-local pivot update. The one-bead updates move changes the position of a single randomly selected bead without breaking the chain. The two-bead update works similarly. The non-local pivot move chose one bead at random and pivots all beads coming after it in the sequence, through a rigid-body rotation or reflection. The pivot move speeds up the simulation of long flexible chain, which evolve much more slowly if only one- and two local moves are used. The local moves are needed if the chain is in a compact folded state, because pivot moves are unlikely to be accepted in this case. At each temperature, 10^5 sweeps were performed, with one sweep consisting of $N-1$ one-bead moves, $N-2$ two-bead moves and one pivot move. The chains were not confined to a finite-sized grid.

The simulations were run on a standard desktop computer. For $N = 30$, each QUBO SA run required 21 CPU-core-seconds, whereas each explicit-chain SA run required 5 CPU-core-seconds. To gather statistics, for each sequence, we performed 1000 runs with each method, using different random number seeds.

5.2 Hybrid Quantum-Classical Computations

We conducted hybrid quantum-classical computations for HP chains with up to 64 beads, using D-Wave's Leap hybrid sampler service and a D-Wave Advantage quantum annealer. All sequences were folded on a 10^2 grid. For $N = 64$ ($N = 30$), each hybrid run required 8 (4) seconds. To gather statistics, 100 runs were performed for each sequence, except for the $N = 64$ chain where 200 runs were performed.

5.3 QPU Computations

For the pure QPU computations, we used the `DWaveCliqueSampler`, designed for dense binary quadratic models. All the computations used a chain strength between 1 and 7.5 and the annealing time was set to $\tau = 2000\,\mu$s, its maximum allowed value. The number of output reads per run, `num_reads`, was set to 490.

5.4 HP Sequences

As a testbed, we used a selected set of HP sequences with 4–30 beads, all of which are known from exhaustive enumerations to have a unique minimum-energy structure [18, 19]. The sequences are labeled S_N, where N indicates the number of beads, and are shown in Table A1 along with their minimum energies, E_{min}. The corresponding structures are illustrated in Figs. 7, 9 and A1..

For a given $N \leq 30$, the selected sequence S_N is one of those that design the structure with highest designability for that N. The two sequences S_{48} and S_{64} probably do not have a unique minimum-energy structure. They were chosen since they have been widely used to test (novel) classical algorithms [20, 37, 38].

6 Results

Using the spin representation of sec. 4.2, we wish to find minimum-energy structures of given HP sequences by minimizing the total energy $E = E_{\mathrm{HP}} + \sum_i \lambda_i E_i$ (Eq. 4.15) on a quantum annealer. As a first step toward this goal, we investigate the power of the QUBO approach under SA, and how it depends on the Lagrange parameters λ_i. We next do the same using the hybrid quantum-classical solver. Finally, we compare the results by using the QPU annealer only.

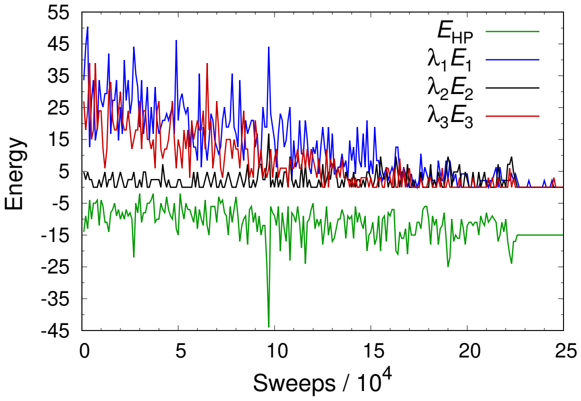

Figure 4: Run-time evolution of the HP energy E_{HP} and the constraint energies E_1, E_2 and E_3 in a QUBO SA folding simulation for the 30-bead sequence S_{30} (Table A1) on 10^2 grid, with $\vec{\lambda} = (2.1, 2.4, 3.0)$.

6.1 Simulated Annealing with QUBO Encoding

In the binary model presented in sec. 4.2, the E_{HP} energy can become substantially lower than it is in any proper chain state. For this QUBO approach to work, it is therefore essential that the λ_i parameters that force the solutions to be “legal” are sufficiently large. On the other hand, by choosing large λ_i values, there is a risk of making the energy landscape rugged, and therefore the dynamics potentially slow. Hence, the λ_i parameters should be neither too large nor too small.

To gain insight into the behavior of the binary model and its dependence on the λ_i parameters, a set of classical Monte Carlo-based SA runs (Sec. 5.1) were conducted, using the HP sequences S_{18}–S_{30} in Table A1. The runs had a fixed length and were deemed successful if the final state corresponded to the known minimum-energy structure of the given HP sequence. As expected, in order to have an acceptable hit rate, it was necessary to choose the λ_i parameters with some care. Nevertheless, without excessive fine-tuning, it was possible to find a single set of parameters, $\vec{\lambda} = (2.1, 2.4, 3.0)$, that gave a hit rate $\gtrsim$0.1 for all the sequences S_{18}–S_{30} (see below). We refrained from attempting any further optimization of the parameters, as the optimal values need not be the same on a quantum annealer. The optimal parameters would, of course, also depend on both HP sequence and grid size.

Fig. 4 shows the run-time evolution of the four different energy terms in one of 1000 QUBO SA runs for the sequence S_{30}. At the end of the run, all the three constraint energies E_i are zero, while E_{HP} takes its known minimum value for an HP chain with this sequence ($E_{\mathrm{min}} = -15$). Hence, the final spin configuration corresponds to the S_{30} ground state in the HP model. The fraction of runs ending in the ground state, the hit rate, was 0.226 ± 0.013. The remaining runs ended in spin configurations that either did

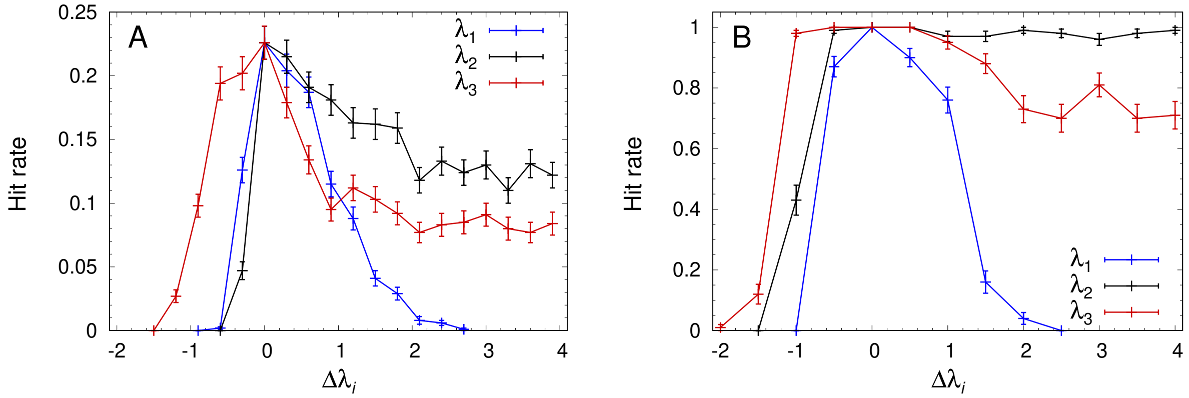

Figure 5: Parameter dependence of the fraction of correct solutions (hit rate) in the vicinity of a reference point $\vec{\lambda}^*$, when using QUBO SA and hybrid quantum-classical computation to search for the ground state of the S_{30} sequence (Table A1) on a 10^2 grid. The hit rate is plotted against $\Delta\lambda_i = \lambda_i - \lambda_i^*$, keeping $\lambda_j = \lambda_j^*$ for $j \neq i$. Lines are drawn to guide the eye. (A) QUBO SA with $\vec{\lambda} = (2.1, 2.4, 3.0)$. (B) Hybrid quantum-classical computations with $\vec{\lambda} = (2.0, 3.0, 3.0)$. Note the difference in scale between the two panels, reflecting the difference in performance as shown in Fig. 6.

not correspond to a proper chain, or corresponded to a structure with $E_{\text{HP}} > E_{\text{min}}$. In the beginning of the runs, the spin system undergoes a rapid relaxation, which brings the energies from initial values $E_{\text{HP}} \sim -10^3$ and $\lambda_1 E_1 + \lambda_2 E_2 + \lambda_3 E_3 \sim 10^5$ to the plotted range before the first measurement is taken (after 10^3 sweeps). It is interesting to note that among the three constraint energies, E_1 and E_3 tend to relax much more slowly than E_2, as is the case in Fig. 4. Note also that the HP energy takes values $E_{\text{HP}} < E_{\text{min}}$ many times during the course of the run. Such values can occur only when at least one constraint is broken.

Based on a limited set of preliminary runs, the parameters $\vec{\lambda} = \vec{\lambda}^* = (2.1, 2.4, 3.0)$ were chosen for this QUBO SA run (Fig. 4). Figure 5a shows the parameter dependence of the hit rate in the vicinity of $\vec{\lambda}^*$, when changing one λ_i at a time. For all three parameters, the hit rate stays tiny until a threshold is passed, followed by a steep increase to the maximum observed hit rate, for $\lambda_i = \lambda_i^*$. When further increasing λ_i beyond λ_i^*, the hit rate decays, most likely due to an increasingly rugged energy landscape. This decay leads to an upper limit on the parameter λ_1, beyond which the hit rate is impractically small. By contrast, the hit rate stays significant even for λ_2 and λ_3 values much larger than those in Fig. 5a. In fact, setting $\lambda_2 = 100$ or $\lambda_3 = 100$, we still obtained hit rates of 0.132 ± 0.011 and 0.074 ± 0.008, respectively. Hence, overall the parameter sensitivity is low, although λ_1 must be chosen with some care.

The fact that the λ_1 dependence has a different shape than the dependencies on λ_2 and λ_3 can be, at least in part, understood. With the single-spin updates employed, the system

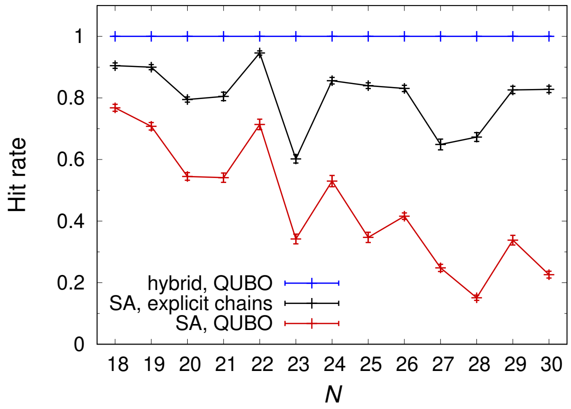

Figure 6: Fraction of correct solutions (hit rate) when searching for the ground state of HP chains with $18 \leq N \leq 30$ beads by using the D-Wave hybrid solver with QUBO encoding (blue), SA with explicit chains (black) and SA with QUBO encoding (red). The HP sequences studied can be found in Table A1. The parameters $\vec{\lambda}$ were set to $(2.1, 2.4, 3.0)$ in the QUBO SA runs, and to $(2.0, 3.0, 3.0)$ when using the hybrid solver. All QUBO-based results were obtained using a 10^2 grid.

cannot move from one chain-like configuration to another, both with $E_1 = E_2 = E_3 = 0$, without visiting intermediate non-chain configurations with $E_1 > 0$. By contrast, E_2 and E_3 may stay zero during such a move. This observation suggests that the energy landscape becomes rugged for large λ_1, but not necessarily so for large λ_2 or λ_3.

To explore how the performance of the QUBO SA approach depends on chain length, we conducted calculations for all the HP sequences S_{18}–S_{30} in Table A1, using $\vec{\lambda} = \vec{\lambda}^*$. As expected, the measured hit rates show a decreasing trend with increasing N (Fig. 6). However, the decrease is not monotonous, indicating that the hit rate is sequence-dependent and not a simple function of N.

For comparison, we also carried out a set of direct SA minimizations of E_{HP} based on conventional explicit-chain Monte Carlo methods (Fig. 6). Despite being faster, the hit rate is higher in these runs than it is with QUBO-based SA. However, the difference in hit rate is modest given that state space in comparison is tiny with explicit chains. Note the similarities in shape between the hit rates obtained from these two unrelated sets of SA calculations. These similarities suggest that some target structures are relatively easy or difficult to find, independent of the method employed.

6.2 Hybrid Quantum-Classical Computations

A promising approach that D-Wave offers is provided by hybrid quantum-classical methods, by which large systems can be studied. To assess the power of this approach, we conducted hybrid computations for all the HP sequences studied in sec. 6.1, S_{18}–S_{30} (Table A1). We additionally included two longer sequences [20], which have been extensively used as testbeds for various (classical) methods.

As in the SA case, with the hybrid solver, a rough search was sufficient in order to find a single $\vec{\lambda}$, $\vec{\lambda}^* = (2.0, 3.0, 3.0)$, for which all the sequences S_{18}–S_{30} could be correctly folded on a 10^2 grid. Figure 5b shows the parameter dependence of the hit rate near $\vec{\lambda}^*$ when using the hybrid solver. Compared to QUBO SA (Fig. 5a), the measured hit rates are significantly higher with the hybrid solver. At the same time, the shapes of the λ_i dependencies are similar in both cases. In particular, in both cases, the hit rate is more sensitive to changes in λ_1 than to changes in λ_2 or λ_3. Similarly to the SA case, the hit rate stays significant for values of λ_2 or λ_3 that are larger than those in Fig. 5b. In particular, when setting $\lambda_2 = 100$ or $\lambda_3 = 100$, we obtained hit rates of 0.980 ± 0.014 and 0.540 ± 0.05, respectively. Overall, the parameter sensitivity of the model is lower with the hybrid solver than with QUBO SA.

When comparing hit rates from our hybrid and QUBO SA runs for the sequences S_{18}–S_{30}, we find that it is consistently highest in the hybrid case (Fig. 6). In fact, the hit rate is one across this entire set of sequences for the hybrid solver. It is important to note that when folding the sequences S_{18}–S_{30}, the hybrid solver did not always make use of the QPU. The fraction of runs that used the QPU increased with N and was above one half for $N > 21$. Still, the precise contribution of the QPU to the final results is hard to judge since the details of the inner-workings of the hybrid solver are not publicly available information (see sec. 3.2). Nevertheless, the impressive results obtained for these sequences motivated us to also test the hybrid solver on two significantly longer sequences, namely S_{48} and S_{64} (Table A1) with 48 and 64 beads, respectively.

For these two sequences, exact results are not available, but both belong to a set of HP sequences that have been widely used to test novel (classical) algorithms [20]. The lowest known energies are $E_{\mathrm{HP}} = -23$ for S_{48} [38] and $E_{\mathrm{HP}} = -42$ for S_{64} [37]. We performed hybrid computations for both sequences on a 10^2 grid. In order to obtain good results, the λ_i parameters had to be adjusted. For S_{48}, the lowest known energy, $E_{\mathrm{HP}} = -23$, was recovered in 10 of 100 hybrid runs, using $\vec{\lambda} = (2.0, 3.5, 3.0)$. One of these 10 structures is shown in Fig. 7a. For S_{64}, with $\vec{\lambda} = (3.0, 4.0, 4.0)$, a structure with the lowest known energy, $E_{\mathrm{HP}} = -42$, was obtained in 1 of 200 hybrid runs. This folded structure is depicted in Fig. 7b. Although the hit rate is low, particularly for S_{64}, it is encouraging that the hybrid solver is able to locate these complex low-energy structures.

Lastly, it is interesting to note that both structures in Fig. 7 exhibit a hydrophobic (H) core and a polar (P) shell. This is in agreement with the observation that most low-energy configurations contain a hydrophobic core. In fact, the structure in Fig. 7a has a perfect square of H beads, while the structure in Fig. 7b almost has a rectangle. This hints at the energies being very close, if not equal, to the global minimum-energies.

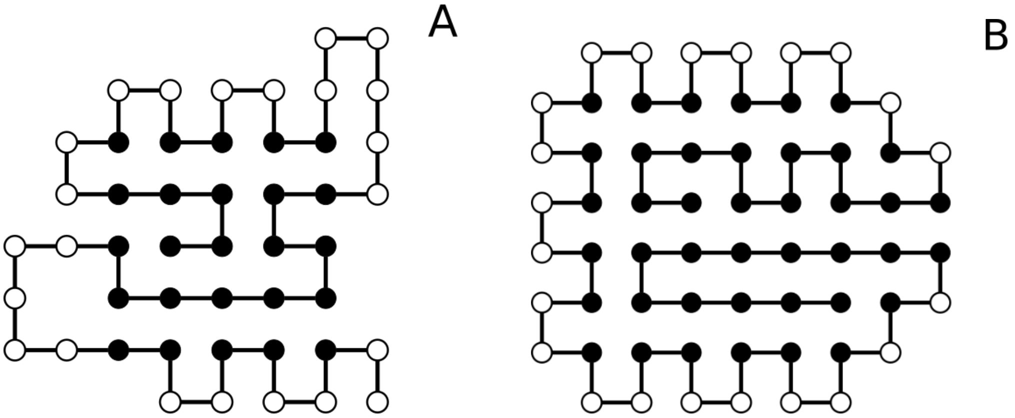

Figure 7: Low-energy structures from hybrid classical-quantum computations for the sequences S_{48} and S_{64} (Table A1) obtained on a 10^2 grid. Filled and open symbols indicate H and P beads, respectively for both structures. (A) S_{48}-structure with energy $E_{\mathrm{HP}} = -23$, obtained with parameters $\vec{\lambda} = (2.0, 3.5, 3.0)$. (B) S_{64}-structure with energy $E_{\mathrm{HP}} = -42$, obtained with parameters $\vec{\lambda} = (3.0, 4.0, 4.0)$).

6.3 Pure QPU Computations

The QUBO problem that we wish to solve for finding minimum-energy HP structures contains $\approx NL^2/2$ logical qubits (variables). Moreover, the system is almost fully connected, implying that its embedding into the QPU topology requires a significant amount of additional qubits. Therefore, pure QPU computation is effectively limited to relatively short HP chains.

To explore how the performance of the pure QPU approach depends on system size, we conducted computations for the six sequences S_4, S_6, S_7, S_8, S_9 and S_{10} (Table A1) for various grid sizes. Figure 8a shows the fraction of all annealing cycles that recovered the known minimum-energy structure, for these systems, plotted against the number of physical qubits required. The parameters λ_i and the annealing time were the same for all systems, whereas the chain strength was chosen individually for each system, for best performance (among the values 1.0, 1.5,…, 4.5, 5.0). Albeit with some scatter, the hit rate shows a roughly exponential decay with system size, as expected the decay is due to stems from integrated control errors (eq. 3.10) [30]. This system-size dependence is more severe than the polynomial scaling of the number of qubits required, and remedies are being explored [30].

The embeddings of our almost fully connected problem into the QPU topology using the `DWaveCliqueSampler`, require the use of chains of physical qubits. As a result, when the problem size increases, the number of physical qubits differs from the number of logical qubits. In Fig. 8b, the number of physical qubits is plotted against the number of logical qubits for the same systems as in Fig. 8a. We find that the data are well described by a

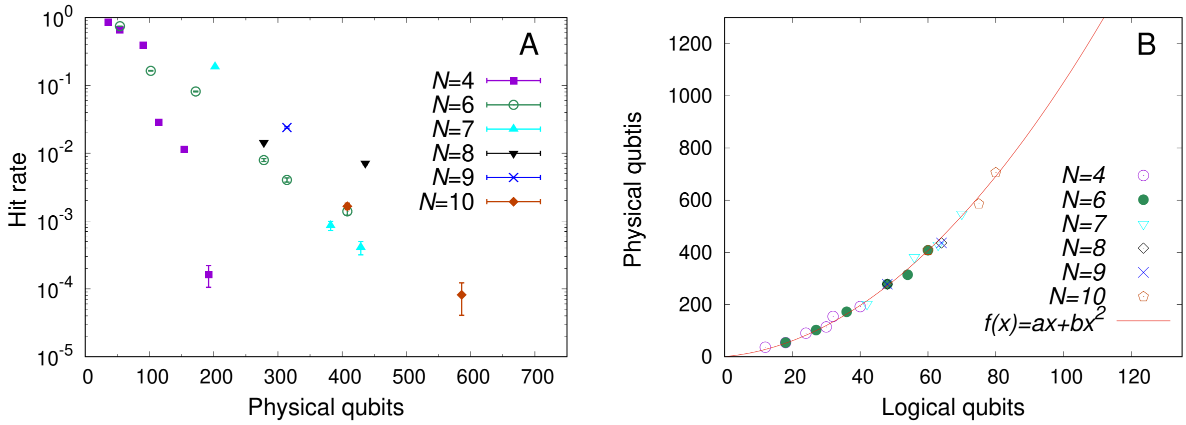

Figure 8: Pure QPU computations. (A) The hit rate on a logarithmic scale against the number of physical qubits used for the sequences S_4, S_6, S_7, S_8, S_9 and S_{10} for various grid sizes. For every choice of sequence and grid size studied, we conducted 100 runs with 490 annealing cycles each, using $\vec{\lambda} = (1.0, 4.0, 1.5)$. Statistical errors are in many cases comparable with or smaller than the symbol sizes. (B) The number of physical qubits against the number of logical qubits for the sequences S_4, S_6, S_7, S_8, S_9 and S_{10} with various grid sizes. The curve shows the fitted function $f(x) = ax+bx^2$, with $a = 1.85$ and $b = 0.093$. The embeddings into the QPU topology was generated using `DWaveCliqueSampler`.

quadratic function $f(x) = ax + bx^2$, with $a = 1.85$ and $b = 0.093$.

The longest sequence whose ground state was successfully recovered in our pure QPU computations was S_{14} with 14 beads (Table A1). This sequence, whose minimum-energy structure can be seen in Fig. 9, was studied using a 4^2 grid, which required 112 logical and 1214 physical qubits. The chain strength was set to 7.5. To our knowledge, this is the largest protein successfully folded using a quantum computer. It should be possible to simulate larger chains using this encoding, as only 1214 of the +5000 qubits were used. However, this will most likely prove to be cumbersome due to the aforementioned drastic decay of the hit rate. In fact, the ground state of S_{14} was only recovered in one of a total of 100 × 490 annealing cycles. This number of cycles is larger than the number of states available to a chain with 14 beads on a 4^2 grid (counting all conformations of homopolymers of size 14 on a 4^2 grid yields 416). On the other hand, it is tiny compared to the $2^{112} \approx 5.2 \times 10^{33}$ states of the binary system, the vast majority of which do not correspond to proper chain configurations.

It is possible that the pure QPU results can improved by further tuning of the simulation parameters. However, at present, we conclude that pure QPU computation cannot match classical SA or the hybrid quantum-classical approach (secs. 6.1, 6.2).

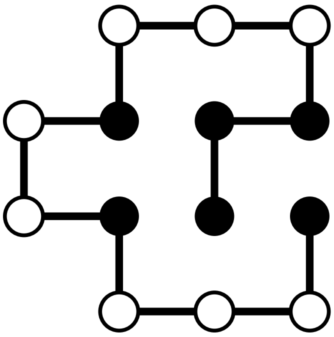

Figure 9: The minimum-energy structure for the sequence S_{14}, which was successfully recovered on a 4^2 grid. This structure was obtained using the parameters $\vec{\lambda} = (2.0, 7.0, 4.0)$ and a chain strength of 7.5. Filled and open symbols indicate H and P beads, respectively. The sequences can be found in Table A1.

7 Conclusion and Outlook

In this project, we have developed a novel spin system representation of HP lattice proteins, for structure prediction by energy minimization on a quantum annealer. This approach uses an energy function that contains three auxiliary penalty terms, whose role is to ensure that the generated spin configurations correspond to proper chains. A configuration is chain-like if and only if all three penalty terms vanish.

To make sure that our energy function gives the correct structures upon minimization, we tested it using classical SA, with the HP sequences S_{18}–S_{30} in Table A1 as a testbed. We also wanted to gauge how the performance of this approach depends on the λ_i parameters. All three parameters have to be sufficiently large in order to obtain chain-like configurations. However, if λ_1 is chosen too large, the energy landscape becomes rugged, leading to a poor success rate. In contrast, λ_2 and λ_3 can be increased without any drastic drop in hit rate. Thus, the approach is robust to changes in the λ_i parameters, especially for λ_2 and λ_3.

Having verified that this mapping can be used to determine structure by classical SA, we moved on to hybrid quantum-classical computations. For the sequences S_{18}–S_{30}, the hybrid approach recovered the minimum-energy structure in 100 % of the runs. In fact, it outperformed classical SA with the same encoding, both in terms of hit rate and speed. The robustness to changes in λ_i parameters was found to be even better with the hybrid method than it was with SA. The strong performance of the hybrid approach motivated us to attempt to fold the two longer sequences S_{48} and S_{64}, for both of which the lowest-known energy was recovered. Especially for S_{64}, this energy level is non-trivial to find, as shown by the fact that some advanced classical algorithms failed to do so.

Finally, we performed pure quantum annealing computations based on this mapping. When doing this, we gauged the hit rate of the method for the sequences S_4–S_{10} for various grid sizes. The hit rate showed a steep decrease with increasing system size, implying that pure QPU computations cannot match classical methods yet. We attributed this decrease to integrated control errors (eq. 3.10) [30]. However, we were able to recover the minimum-energy structure for S_{14} on a 4^2 grid. To our knowledge, this is the largest lattice protein that has been successfully folded using a quantum computer.

All calculations in this project focused on the 2D HP model. It should therefore be pointed out that the approach can be extended in at least two ways. Firstly, it can handle models with larger alphabets, such as the 20-letter Miyazawa-Jernigan model [39], as it amounts to simply changing the $C(h_f, h_{f'})$ parameters in eq. 4.16. Secondly, although the checkerboard division into odd and even sites may have to be modified or abandoned, the approach can be used on an arbitrary graph. Thus, it can handle three-dimensional grids or any other graph of relevance.

Further, we hope that this approach can be useful for gate-based quantum computations as well. This could potentially be in the form of a quantum variational algorithm or a quantum search algorithm.

Acknowledgments

I would like to thank Professor Anders Irbäck for his excellent guidance. I would also like to thank Carsten Peterson and Sandipan Mohanty for discussion and guidance. I have enjoyed fruitful discussions with theory group members at the Wallenberg Centre for Quantum Technology at Chalmers University. In particular, I would like to thank Hanna Linn for feedback on the code. Finally, it is a pleasure to gratefully acknowledge the Jülich Supercomputing Centre (https://www.fz-juelich.de/ias/jsc) for supporting this project by providing computing time on the D-Wave Advantage™ System JUPSI through the Jülich UNified Infrastructure for Quantum computing (JUNIQ).

Appendix

Table A1: The HP sequences studied. The sequences are labeled S_N, where N indicates the number of beads. All S_N with $N \leq 30$ have a known, unique minimum-energy structure [18, 19]. The minimum energy is denoted by E_{min}. For all $N \leq 30$, the sequence S_N is chosen among those having the most highly designable structure for this N as its unique minimum-energy structure. For the additional and longer sequences S_{48} and S_{64}, the ground states are unknown. Here, the E_{min} values, marked with an asterisk, are the currently lowest known energies, found with classical methods [37, 38]. Low-energy structures for all the sequences studied can be found in Figs. 7 (S_{64} and S_{48}), 9 (S_{14}), and A1 (all other S_N).

Name	Sequence	E_{min}
S_4	HPPH	-1
S_6	HPPHPH	-2
S_7	PHPPHPH	-2
S_8	HPHPHPPH	-3
S_9	HHPPHPPHP	-3
S_{14}	HHHPPPHPPHPPPH	-5
S_{18}	HHHPPHPPHPHPPHPHPH	-9
S_{19}	PHPHPHPPHPHPPHPPHHH	-9
S_{20}	HPHPHPPHPHPPHPPPPHHH	-9
S_{21}	PHHPPHPHPPHPHPPHPPHHH	-10
S_{22}	HPPHPPHPHPPHPHPPHPPHHH	-11
S_{23}	PPHHHHPPHPPHPHPPHPHPPHP	-10
S_{24}	HPPPPHPPHPHPPHPHPPHPPHHH	-11
S_{25}	PHPHPHPHPPHPHPHPPHPPHHHHH	-13
S_{26}	HHHHPPHHPPHPHPPHPHPPHHPPHH	-14
S_{27}	PHPHPHPHPPHPHPHPPHPPPPHHHHH	-13
S_{28}	PPHHHPPHPPHPHPHPPHPHPPHPPHHH	-13
S_{29}	PHPHPHPPHHPPHPHPPHPPHHHHPPHHH	-15
S_{30}	PPHHHHPPHPPHPHPPHHPPHPHPHPPHHH	-15
S_{48}	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH	$-23*$
S_{64}	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHPPHHPPHHPPHPHPHHHHHHHHHHHH	$-42*$

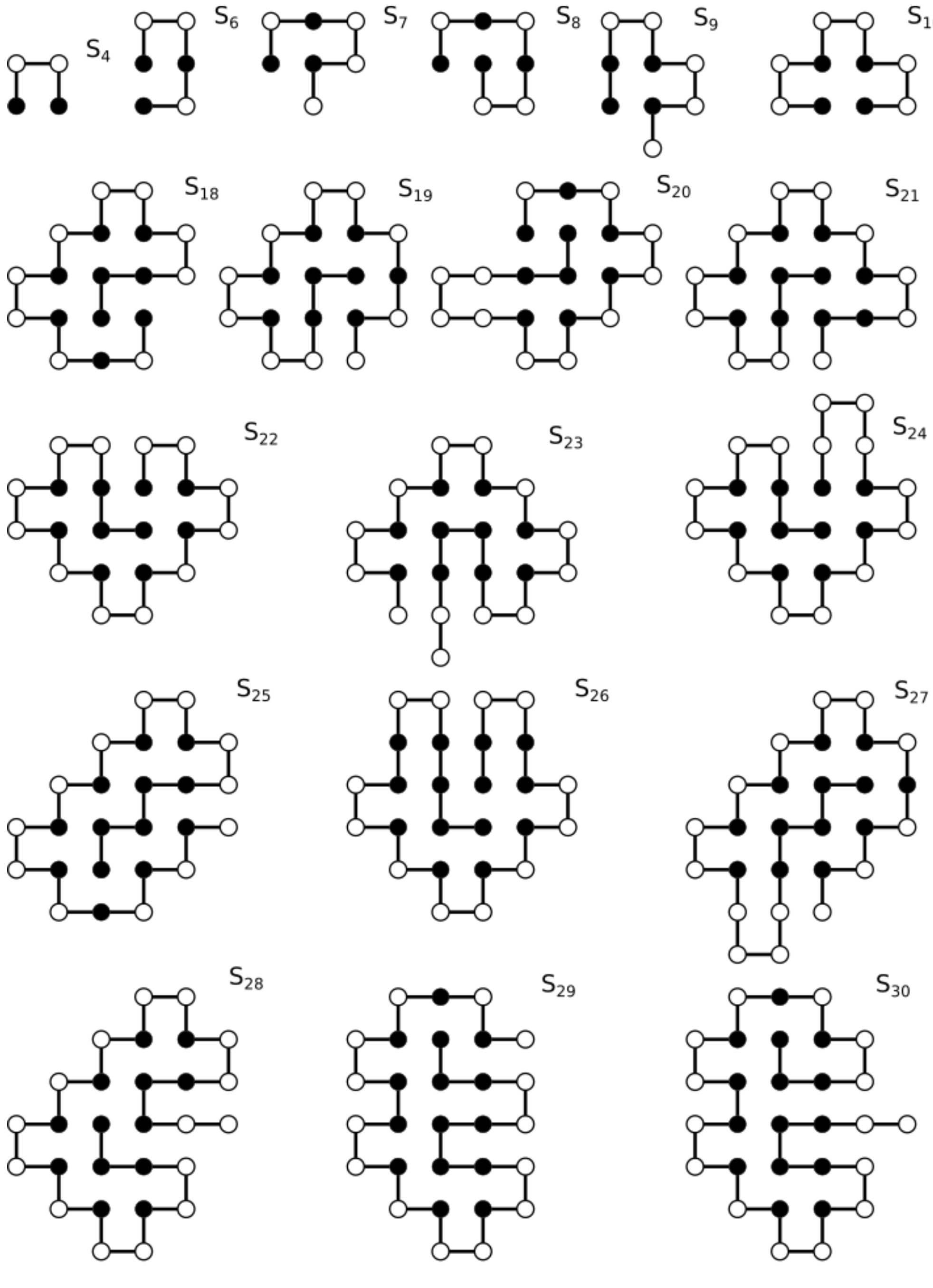

Figure A1: Ground states for all the sequences S_N in Table A1 with $N \leq 30$ [18,19] except S_{14} (whose ground state can be found in Fig. 9). Low-energy structures for the two final sequences in Table A1, S_{64} and S_{48}, can be found in Fig. 7.

References

[1] Ferrini G, Anton Frisk Kockum LGA, Vikstål P. Advanced Quantum Algorithms; 2020. Lecture notes from the course Advanced Quantum Algorithms, given at Wallenberg Centre of Quantum Technology at Chalmers University of Technology in autumn of 2020.

[2] Nielsen MA, Chuang IL. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press; 2010.

[3] Johnson MW, Amin MHS, Gildert S, Lanting T, Hamze F, Dickson N, et al. Quantum annealing with manufactured spins. Nature. 2011;7346:194-8.

[4] Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, et al. Quantum supremacy using a programmable superconducting processor. Nature. 2019;574:505-10.

[5] Lucas A. Ising formulations of many NP problems. Front Phys. 2014;2:1-15.

[6] Ushijima-Mwesigwa H, Negre CFA, Mniszewski SM. Graph Partitioning Using Quantum Annealing on the D-Wave System. In: Proceedings of the Second International Workshop on Post Moores Era Supercomputing. PMES'17. New York, NY, USA: Association for Computing Machinery; 2017. p. 22–29. Available from: https://doi.org/10.1145/3149526.3149531.

[7] Peterson C, Anderson JR. Neural networks and NP-complete problems; a performance study of the graph bisectioning problem. Complex Syst. 1988;2:59-89.

[8] Hopfield JJ, Tank DW. Neural computation of decisions in optimization problems. Biol Cybern. 1985;52:141-52.

[9] Perdomo A, Truncik C, Tubert-Brohman I, Rose G, Aspuru-Guzik A. Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. Phys Rev A. 2008;78:012320.

[10] Lau KF, Dill KA. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules. 1989;22:3986-97.

[11] Perdomo-Ortiz A, Dickson N, Drew-Brook M, Rose G, Aspuru-Guzik A. Finding low-energy conformations of lattice protein models by quantum annealing. Sci Rep. 2012;2:248.

[12] Robert A, Barkoutsos PK, Woerner S, Tavernelli I. Resource-efficient quantum algorithm for protein folding. Npj Quantum Inf. 2021;7:38.

[13] Outeiral C, Morris GM, Shi J, Strahm M, Benjamin SC, Deane CM. Investigating the potential for a limited quantum speedup on protein lattice problems. New J Phys. 2021;23:103030.

[14] Micheletti C, Hauke P, Faccioli P. Polymer physics by quantum computing. Phys Rev Lett. 2021;127:080501.

[15] Babbush R, Perdomo-Ortiz A, O'Gorman B, Macready W, Aspuru-Guzik A. Construction of energy functions for lattice heteropolymer models: efficient encodings for constraint satisfaction programming and quantum annealing. Adv Chem Phys. 2014;155:201-44.

[16] Kirkpatrick S, Gelatt CD, Vecchi M. Optimization by simulated annealing. Science. 2014;220:671-80.

[17] McGeocha C, Farré P. The D-Wave Advantage System: An Overview; 2020.

[18] Irbäck A, Troein C. Enumerating designing sequences in the HP model. J Biol Phys. 2002;28:1-15.

[19] Holzgräfe C, Irbäck A, Troein C. Mutation-induced fold switching among lattice proteins. J Chem Phys. 2011;135:195101.

[20] Unger R, Moult J. Genetic algorithms for protein folding simulations. J Mol Biol. 1993;231:75-81.

[21] Willsch M, Willsch D, Michielsen K. Lecture Notes: Programming Quantum Computers; 2022.

[22] Albash T, Lidar DA. Adiabatic quantum computation. Rev Mod Phys. 2018 Jan;90:015002. Available from: https://link.aps.org/doi/10.1103/RevModPhys.90.015002.

[23] Harris R, Lanting T, Berkley AJ, Johansson J, Johnson MW, Bunyk P, et al. Compound Josephson-junction coupler for flux qubits with minimal crosstalk. Phys Rev B. 2009 Aug;80:052506. Available from: https://link.aps.org/doi/10.1103/PhysRevB.80.052506.

[24] Harris R, Johansson J, Berkley AJ, Johnson MW, Lanting T, Han S, et al. Experimental demonstration of a robust and scalable flux qubit. Phys Rev B. 2010 Apr;81:134510. Available from: https://link.aps.org/doi/10.1103/PhysRevB.81.134510.

[25] Harris R, Johnson MW, Lanting T, Berkley AJ, Johansson J, Bunyk P, et al. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. Phys Rev B. 2010 Jul;82:024511. Available from: https://link.aps.org/doi/10.1103/PhysRevB.82.024511.

[26] Raussendorf R, Briegel HJ. Quantum computing via measurements only. arXiv; 2000. Available from: https://arxiv.org/abs/quant-ph/0010033.

[27] Bartolucci S, Birchall P, Bombin H, Cable H, Dawson C, Gimeno-Segovia M, et al.. Fusion-based quantum computation. arXiv; 2021. Available from: https://arxiv.org/abs/2101.09310.

[28] D-Wave. Other Error Sources;. Accessed: 2022-03-11. https://docs.dwavesys.com/docs/latest/c_qpu_errors.html.

[29] D-Wave. Error Sources for Problem Representation;. Accessed: 2022-03-11. https://docs.dwavesys.com/docs/latest/c_qpu_ice.html#gaussian-systematic-j.

[30] Pearson A, Mishra A, Hen I, Lidar DA. Analog errors in quantum annealing: doom and hope. Npj Quantum Inf. 2019;5:107.

[31] Date P, Patton R, Schuman C, Potok T. Efficiently embedding QUBO problems on adiabatic quantum computers. Quantum Information Processing. 2019 Mar;18(4):117.

[32] D-Wave. Problem-Solving Handbook;. Accessed: 2022-03-11. https://docs.dwavesys.com/docs/latest/doc_handbook.html.

[33] D-Wave. D-Wave Ocean Software Documentation;. Accessed: 2022-03-11. https://docs.ocean.dwavesys.com/en/stable/.

[34] D-Wave. D-Wave Ocean SDK;. Accessed: 2022-03-11. https://github.com/dwavesystems/dwave-ocean-sdk.

[35] D-Wave. Samplers;. Accessed: 2022-03-11. https://docs.ocean.dwavesys.com/projects/system/en/stable/reference/samplers.html?highlight=leaphybridsampler#leaphybridsampler.

[36] Catherine McGeoch PF, Bernoudy W. D-Wave Hybrid Solver Service + Advantage: Technology Update; 2020.

[37] Bastolla U, Frauenkron H, Gerstner E, Grassberger P, Nadler W. Testing a new Monte Carlo algorithm for protein folding. Protein Eng. 1998;32:52-66.

[38] Liang F, Wong WH. Evolutionary Monte Carlo for protein folding simulations. J Chem Phys. 2001;115:3374-80.

[39] Miyazawa S, Jernigan RL. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. J Mol Biol. 1996;256:623-44.