Problem Solving with Hopfield Networks and Adiabatic Quantum Computing

Christian Bauckhage
Computer Science
University of Bonn
Bonn, Germany

Ramses Sanchez
B-IT
University of Bonn
Bonn, Germany

Rafet Sifa
Fraunhofer Center for Machine Learning
Fraunhofer IAIS
St. Augustin, Germany

***Abstract*—Our goal with this paper is to elucidate the close connection between Hopfield networks and adiabatic quantum computing. Focusing on their use in problem solving, we point out that the energy functions minimized by Hopfield networks are essentially identical to those minimized by adiabatic quantum computers. To practically illustrate this, we consider a simple textbook problem, namely the k-rooks problem, and discuss how to set it up for solution via a Hopfield network or adiabatic quantum computing.**

***Index Terms*—problem solving, neural networks, quantum computing**

I. Introduction

Quantum computing exploits quantum mechanical phenomena such as superposition or entanglement for information processing. Since this promises levels of efficiency unreachable by digital computers and since noisy intermediate-scale quantum (NISQ) computers are now a technical and economical reality [1]–[3], it appears worthwhile to consider their use in artificial intelligence and machine learning. Indeed, a quickly growing body of work suggests that quantum computers can efficiently solve a wide range of optimization problems commonly encountered in these fields [4]–[10].

However, as of this writing, corresponding reports are still largely confined to the physics literature and have not yet received widespread attention in the computational intelligence community. Likely reasons are that quantum mechanics is seemingly abstract and hard to fathom because its effects do not manifest in our daily lives. Also, the mathematics behind quantum mechanics is often not taught in common computer science curricula and the corresponding mathematical notation needs getting used to as well.

Yet, problem solving on a quantum computer does not necessarily require a deeper understanding of quantum mechanics just as problem solving on a digital computer does not require a deeper understanding of Boolean algebra. What is required, though, are different perspectives on problem solving and it is interesting to note that one such perspective is actually well known to neurocomputing experts.

Hence, our main goal with this paper is to point out that certain quantum computing techniques are actually not far removed from certain neural network based approaches. In particular, we intend to emphasize the close connection between Hopfield networks and the paradigm of adiabatic quantum computing (AQC).

To this end, we first recall the basic theory behind Hopfield networks and their use in problem solving (section II). We then recall basic premises of quantum computing (section III) and finally review the ideas behind AQC (section IV). This will reveal that everything a Hopfield network can do, an adiabatic quantum computer can do, too, but arguably better.

II. Hopfield Networks for Problem Solving

Recall that a Hopfield network is a recurrent neural network of n interconnected neurons $s_1, s_2, \ldots, s_n$ each of which is a bipolar threshold unit

$$s_i = \operatorname{sign}\big(\boldsymbol{w}_i^\mathsf{T}\boldsymbol{s} - \theta_i\big) \tag{1}$$

where the vector $\boldsymbol{s} \in \{-1, +1\}^n$ denotes the global state of the whole network.

If the $n \times n$ weight matrix $\boldsymbol{W}$ of a Hopfield network is symmetric and hollow, i.e. has a diagonal of all 0s, and if the network's neurons update asynchronously, then the energy

$$H(\boldsymbol{s}_t) = -\tfrac{1}{2}\,\boldsymbol{s}_t^\mathsf{T}\boldsymbol{W}\boldsymbol{s}_t + \boldsymbol{\theta}^\mathsf{T}\boldsymbol{s}_t \tag{2}$$

of the network at time t can never increase. As there are only finitely many, namely 2^n distinct states for the network to be in, this is to say that it reaches a (local) energy minimum after finitely many updates [11].

Hopfield networks are most comonly studied in the context of associative memories for pattern retrieval; however, they also allow for problem solving [12]. Here, the basic idea is to devise an energy function H whose minimizers

$$\boldsymbol{s}^* = \operatorname*{argmin}_{\boldsymbol{s}\in\{-1,+1\}^n} -\tfrac{1}{2}\,\boldsymbol{s}^\mathsf{T}\boldsymbol{W}\boldsymbol{s} + \boldsymbol{\theta}^\mathsf{T}\boldsymbol{s} \tag{3}$$

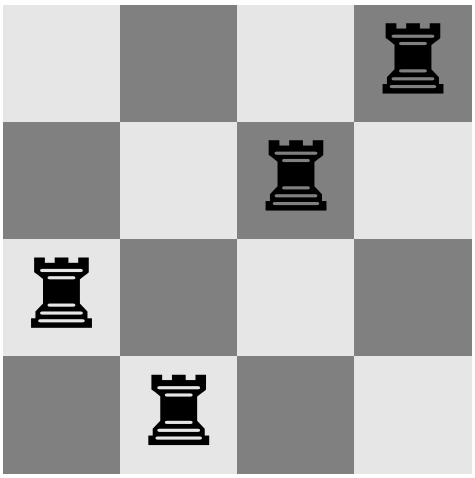
encode solutions to the problem at hand.

Settings where this framework applies include bi-partition and subset selection problems where a given set of n entities needs to be partitioned into two disjoint subsets or a subset of $k < n$ entities needs to be determined. An example for the former is the problem of computing a graph cut; examples for the latter can be found in vector quantization, feature selection, or support vector identification. In both cases, the solution $\boldsymbol{s}^*$ produced by a Hopfield network can be thought of as an indicator vector whose entries $s_i^* \in \{-1, +1\}$ indicate which entities to select or to assign to which subset.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

t	$\boldsymbol{s}_t$	$H(\boldsymbol{s}_t)$
0	+ + + + + + + + + + + + + + + +	224.0
1	+ + + + + + + − + + + + + + + +	184.0
2	+ + + + + + + − + + + + + − + +	144.0
3	+ + + + + + + − + + − + + − + +	104.0
4	− + + + + + + − + + − + + − + +	64.0
5	− + + + + + + − + + − − + − + +	40.0
6	− − + + + + + − + + − − + − + +	16.0
7	− − + + + − + − + + − − + − + +	0.0
8	− − + + + − + − + + − − + − + +	0.0
9	− − + + + − + − + + − − + − + −	−16.0
10	− − + + − − + − + + − − + − + −	−32.0
11	− − + + − − + − + + − − + − + −	−32.0
12	− − + + − − + − + − − − + − + −	−32.0
13	− − + + − − + − + − − − + − + −	−32.0
14	− − + + − − + − + − − − + − + −	−32.0
15	− − + + − − + − + − − − − − + −	−40.0
16	− − + + − − + − + − − − − − + −	−40.0
17	− − + + − − + − + − − − − − + −	−40.0
⋮	⋮	⋮
52	− − − − − − + − + − − − − + − −	−60.0
53	− − − + − − + − + − − − − + − −	−64.0
54	− − − + − − + − + − − − − + − −	−64.0

(a) exemplary evolution of the global state and energy of a Hopfield network that solves the 4-rooks problem



(b) solution found after 52 asynchronous update steps

Fig. 1: Running a Hopfield network of $n = 16$ bipolar neurons s_i which produce outputs in $\{-1, +1\}$ can solve the 4-rooks problem.

A. Didactic Example: The k Rooks Problem

As a didactic example for the problem solving capabilities of Hopfield networks, we consider the k-rooks problem [12]. It asks for a placement of k rooks on a $k \times k$ chessboard such that they do not threaten each other (see Fig. 1(b)) and therefore constitutes a constrained subset selection problem where k out of $n = k^2$ fields need to be identified.

Modeling the configuration or state of the chessboard in terms of a bipolar $k \times k$ matrix $\boldsymbol{S}$ with entries

$$s_{ij} = \begin{cases} +1 & \text{if a rook is placed on field } (i, j) \\ -1 & \text{otherwise,} \end{cases}$$

a valid solution is a configuration of $\boldsymbol{S}$ such that all its row- and column sums amount to $1 \cdot (+1) + (k-1) \cdot (-1) = 2 - k$.

Vectorizing the matrix $\boldsymbol{S} \in \{-1, +1\}^{k \times k}$ by concatenating its rows into a vector $\boldsymbol{s} \in \{-1, +1\}^{k^2}$, a configuration that obeys these constraints can be found by running a Hopfield network.

B. Solution for the Case $k = 4$

For instance, letting $k = 4$ and skipping over a detailed derivation (which can be found in [12]), we may consider the identity matrix $\boldsymbol{I}$, the matrix of all zeros $\boldsymbol{0}$, and the matrix $\boldsymbol{J} = \boldsymbol{1}\boldsymbol{1}^\intercal - \boldsymbol{I}$ all of size 4×4 in order to define a 16×16 weight matrix

$$\boldsymbol{W}_r = -2 \begin{bmatrix} \boldsymbol{J} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{J} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{J} \end{bmatrix}.$$

Together with a 16-dimensional threshold vector

$$\boldsymbol{\theta}_r = -2\,(2 - k)\,\boldsymbol{1}$$

this yields the energy function $H_r = -\frac{1}{2}\,\boldsymbol{s}^\intercal \boldsymbol{W}_r\,\boldsymbol{s} + \boldsymbol{\theta}_r^\intercal \boldsymbol{s}$ of a Hopfield network that would place one rook per row. By the same token, weights

$$\boldsymbol{W}_c = -2 \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$$

and thresholds

$$\boldsymbol{\theta}_c = -2\,(2 - k)\,\boldsymbol{1}$$

yield the energy function $H_c = -\frac{1}{2}\,\boldsymbol{s}^\intercal \boldsymbol{W}_c\,\boldsymbol{s} + \boldsymbol{\theta}_c^\intercal \boldsymbol{s}$ that places one rook per column. Hence, a Hopfield network with energy function

$$H = H_r + H_c \tag{4}$$

would solve our overall problem.

Figure 1 illustrates the evolution of a Hopfield network that minimizes this energy and, in doing so, successfully finds one of the $4! = 24$ possible solutions to the 4-rooks problem.

III. Quantum Computing in a Nutshell

Quantum computers harness the principles of quantum mechanics for information processing. While this offers great computational power, it also works fundamentally different than classical digital computing. In this section, we therefore provide a brief introduction to basic terminology and concepts of quantum computing.

On a digital computer, the basic unit of information is a bit and the mathematics that governs the behavior of collections of bits is Boolean algebra. On a quantum computer, the basic unit of information is a qubit (quantum bit) and the mathematics that models the behavior of collections of qubits is complex linear algebra.

While a classical bit is in either one of two states (0 or 1), the basic premise of quantum computing is that a qubit exists in a superposition of two states and collapses to either one once

measured. Canonical examples of physical systems that exhibit this phenomenon include the polarization of a photon (vertical or horizontal) or the spin of an electron (up or down). However, quantum mechanical effects are not necessarily restricted to the subatomic world but may also occur in completely isolated macroscopic systems. One can, for instance, realize superconducting circuits in which electrical currents flow in two directions simultaneously [13], [14].

To mathematically describe the behavior of qubits, they are modeled as unit vectors in a two-dimensional Hilbert space over $\mathbb{C}$. Using the Dirac notation[1], we write a qubit as a linear combination

$$\big|\psi\big\rangle = a_0 \big|0\big\rangle + a_1 \big|1\big\rangle \tag{5}$$

where the coefficients $a_0, a_1 \in \mathbb{C}$ are called the amplitudes of the basis states $|0\rangle$ and $|1\rangle$. Importantly, they obey the normalization condition

$$|a_0|^2 + |a_1|^2 = 1 \tag{6}$$

and are interpreted as follows: if a measurement is performed on qubit $|\psi\rangle$, the probability of finding it in state $|0\rangle$ is $|a_0|^2$ whereas the probability of finding it in state $|1\rangle$ corresponds to $|a_1|^2$.

Note that measurements performed on $|\psi\rangle$ are irreversible operations because they constitute interactions with the outside world and therefore lead to quantum decoherence. This is to say that, once a qubit has collapsed to either one of its basis states, it henceforth behaves like a classical bit.

Operations on qubits that preserve their quantum mechanical nature are called reversible. Mathematically, these are unitary linear operators $U \in SU_2(\mathbb{C})$ for which we have $UU^\dagger = U^\dagger U = I$. Reversible operators can also be written as $U = e^{-iHt/\hbar}$ where H is yet another operator called the Hamiltonian. It corresponds to the total energy of a quantum system in the sense that its spectrum reflects all possible outcomes of measurements of the system's total energy.

Another premise of quantum computing is that qubits can be combined to form qubit registers. While a single qubit $|\psi\rangle$ exists in a superposition of 2 basis states, a quantum register $|\boldsymbol{\psi}\rangle$ of n qubits exists in a superposition of 2^n basis states. Mathematically, this is to say that

$$\big|\boldsymbol{\psi}\big\rangle = \sum_{i=0}^{2^n-1} a_i \, |\boldsymbol{\psi}_i\rangle \tag{7}$$

where the amplitudes obey $\sum_i |a_i|^2 = 1$ and the basis states $|\boldsymbol{\psi}_i\rangle$ are 2^n-dimensional tensor products[2] of single qubit basis states. For example, for a quantum register of size $n = 3$, we would work with the following $2^3 = 8$ basis states

$$|\boldsymbol{\psi}_0\rangle = \big|0\big\rangle \otimes \big|0\big\rangle \otimes \big|0\big\rangle \equiv \big|000\big\rangle \tag{8}$$
$$|\boldsymbol{\psi}_1\rangle = \big|0\big\rangle \otimes \big|0\big\rangle \otimes \big|1\big\rangle \equiv \big|001\big\rangle \tag{9}$$
$$\vdots$$
$$|\boldsymbol{\psi}_7\rangle = \big|1\big\rangle \otimes \big|1\big\rangle \otimes \big|1\big\rangle \equiv \big|111\big\rangle \tag{10}$$

Given these prerequisites, there are two distinct (yet equivalent [15]) quantum computing paradigms, namely quantum gate computing and adiabatic quantum computing.

Quantum gate computing sequences quantum mechanical operators $U_1, U_2, \ldots$ to manipulate quantum registers and thus to perform computations. Well known algorithms within this paradigm include Shor's prime factorization [16] or Grover's search [17].

Adiabatic quantum computing is the paradigm we consider in this paper. It exploits a phenomenon summarized by the adiabatic theorem [18]. It basically states that if a quantum system starts out in the ground state[3] of a Hamiltonian operator which then gradually changes over a period of time, the system will end up in the ground state of the resulting Hamiltonian.

To harness this for information processing and problem solving on an adiabatic quantum computer, one prepares a system in the ground state of a simple, problem independent Hamiltonian and adiabatically evolves it towards a Hamiltonian whose ground state represents a solution to the problem at hand [19].

One of the challenges in adiabatic quantum computing is thus to devise suitable problem Hamiltonians. However, as we will see next, if the problem at hand is a minimization problem that can be tackled by a Hopfield network, this challenge is minor because it is actually easy to construct Hamiltonians for the energy functions of Hopfield networks.

IV. AQC for Problem Solving

Adiabatic quantum computers such as those produced by D-Wave systems [14] determine minimum energy states of what physicists call Ising models [20]. In other words, they are tailored towards solving the following kind of optimization problem

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}\in\{-1,+1\}^n}{\operatorname{argmin}} \; \boldsymbol{s}^\intercal \boldsymbol{Q}\, \boldsymbol{s} + \boldsymbol{q}^\intercal \boldsymbol{s}. \tag{11}$$

Just as for the energy of a Hopfield network in (3), the 2^n bipolar vectors $\boldsymbol{s}$ over which to minimize in (11) represent possible global states of a system of n entities each of which might be in one of two local states (either $+1$ or -1). The coupling matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ models internal interactions whereas the vector $\boldsymbol{q} \in \mathbb{R}^n$ models external influences.

[1] Readers not familiar with the Dirac notation may think of the two basis *kets* $|0\rangle$ and $|1\rangle$ in terms of Euclidean vectors $[1,0]^\intercal$ and $[0,1]^\intercal$, respectively. Yet, any other pair of orthogonal vectors, say $[1,1]^\intercal$ and $[1,-1]^\intercal$, would work as well. One of the many reasons why physicists prefer the Dirac notation is that it allows for great symbolic flexibility. For example the two basis polarizations of a photon could be written as $|\uparrow\rangle$ and $|\rightarrow\rangle$ and the two basis spins of an electron could be written as $|\uparrow\rangle$ and $|\downarrow\rangle$, respectively.

[2] The tensor product of two kets can be thought of as the Kronecker product of the corresponding Euclidean vectors. For instance $|0\rangle \otimes |1\rangle$ can be understood as $[1,0]^\intercal \otimes [0,1]^\intercal = [0,0,1,0]^\intercal$.

[3] The ground state of a quantum system is its lowest energy state. Mathematically, this is the eigenvector corresponding to the smallest eigenvalue of a Hamiltonian H.

A. Didactic Example: The $k = 4$ Rooks Problem

Since the optimization problem in (11) is algebraically identical to the one in (3), we may let

$$\boldsymbol{Q} = -\tfrac{1}{2}\big(\boldsymbol{W}_r + \boldsymbol{W}_c\big) \tag{12}$$

and

$$\boldsymbol{q} = \boldsymbol{\theta}_r + \boldsymbol{\theta}_c \tag{13}$$

in order to solve the above $k = 4$ rooks problem via adiabatic quantum computing.

Given (12) and (13), we consider a *time dependent* system of $n = k^2 = 16$ qubits

$$\big|\boldsymbol{\psi}(t)\big\rangle = \sum_{i=0}^{2^n-1} a_i(t)\,\big|\boldsymbol{\psi}_i\big\rangle \tag{14}$$

which is in a superposition of $2^{16} = 65.536$ basis states $|\boldsymbol{\psi}_i\rangle$.

If this system evolves under the influence of a time-dependent Hamiltonian $H(t)$, its behavior is governed by the Schrödinger equation

$$\frac{d}{dt}\,\big|\boldsymbol{\psi}(t)\big\rangle = -i\,H(t)\,\big|\boldsymbol{\psi}(t)\big\rangle \tag{15}$$

where we have set $\hbar = 1$. Using this, we consider a period ranging from $t = 0$ to $t = \tau$ and assume the Hamiltonian in (15) to be a convex combination of two static Hamiltonians

$$H(t) = \big(1 - \tfrac{t}{\tau}\big)\,H_B + \tfrac{t}{\tau} H_P. \tag{16}$$

Here, H_B is the *beginning Hamiltonian* whose ground state is easy to construct and H_P is the *problem Hamiltonian* whose ground states encode the solution to our problem. To set up H_P, we therefore follow standard suggestions [21] and define

$$H_P = \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\,\sigma_z^i\,\sigma_z^j + \sum_{i=1}^{n} q_i\,\sigma_z^i \tag{17}$$

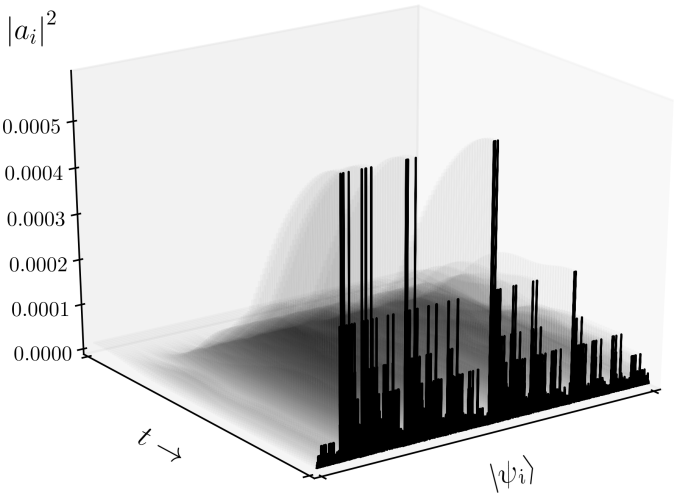
where σ_z^i denotes the Pauli spin matrix σ_z acting on the i-th qubit. Likewise and again following standard suggestions, we choose the beginning Hamiltonian H_B as

$$H_B = -\sum_{i=1}^{n} \sigma_x^i \tag{18}$$

where σ_x^i denotes the corresponding the Pauli spin matrix σ_x.

After letting $|\boldsymbol{\psi}(t)\rangle$ evolve from $|\boldsymbol{\psi}(0)\rangle$ to $|\boldsymbol{\psi}(\tau)\rangle$ where $|\boldsymbol{\psi}(0)\rangle$ corresponds to the ground state of H_B, we may measure the system at the end of this evolution. This causes the qubit system to collapse to one of its 2^{16} basis states and the probability for this state to be $|\boldsymbol{\psi}_i\rangle$ is given by the amplitude $|a_i(\tau)|^2$. However, since the adiabatic evolution was steered towards the problem Hamiltonian H_P, states $|\boldsymbol{\psi}_i\rangle$ that correspond to ground states of H_P are more likely to be found.

The computational efficiency of adiabatic quantum computing will depend on the choice of the duration τ in (16). The optimal choice for this parameter is known to depend on the minimum energy gap between the ground state and the first excited state of $H(t)$. However, since this energy gap implicitly depends on the problem at hand, general closed form expressions for the optimal τ are hard to come by and subject of intensive ongoing research.



(a) amplitude evolution of the 2^{16} basis states

$\|\boldsymbol{\psi}_i\rangle$	$\|a_i\|^2$
$\|0001010010000010\rangle$	0.000594
$\|1000000100100100\rangle$	0.000594
$\|0010010010000001\rangle$	0.000594
$\|0100000100101000\rangle$	0.000594
$\|0100100000100001\rangle$	0.000594
$\|0100001010000001\rangle$	0.000594
$\|1000001000010100\rangle$	0.000594
$\|1000010000100001\rangle$	0.000594
$\|0100000110000010\rangle$	0.000594
$\|0010010000011000\rangle$	0.000594
$\|0010100001000001\rangle$	0.000594
$\|0001001001001000\rangle$	0.000594
$\|0010000110000100\rangle$	0.000594
$\|0001001010000100\rangle$	0.000594
$\|0001100000100100\rangle$	0.000594
$\|1000010000010010\rangle$	0.000594
$\|0001100001000010\rangle$	0.000594
$\|0010100000010100\rangle$	0.000594
$\|0100100000010010\rangle$	0.000594
$\|0100001000011000\rangle$	0.000594
$\|1000000101000010\rangle$	0.000594
$\|0010000101001000\rangle$	0.000594
$\|0001010000101000\rangle$	0.000594
$\|1000001001000001\rangle$	0.000594
$\|0110100010000001\rangle$	0.000285

(b) 25 likeliest final states

Fig. 2: Adiabatic quantum computing for the $k = 4$ rooks problem. The top panel visualizes the evolution of a $n = 16$ qubit systems $|\boldsymbol{\psi}(t)\rangle$. During its evolution over time t, it is in a superposition of $2^{16} = 65.536$ basis states $|\boldsymbol{\psi}_i\rangle$ each representing a potential solutions to the problem. Initially, each possible solution (valid or invalid) is equally likely to be measured; over time, the amplitudes of the various basis states begin to decrease or increase. At the end of the process, 24 basis state have noticeably higher amplitudes $|a_i|^2$ than most other states and are thus more likely to be measured. The table at the bottom ranks the 25 most likely states at the end of the process and shows that these correspond to valid solutions of the problem.

On the other hand, it is known that the gap is inversely proportional to the square root of the number of basis states that have energies close to global minimum [22] and we note that, for the kind of problem considered in this paper, the number of minima is comparatively small ($k! \ll 2^n$ where $n = k^2$). For such problems, $\tau \in O\left(\sqrt{2^n}\right)$ is commonly a good choice for the smallest possible runtime of the adiabatic evolution [23].

Using the *Python* quantum computing toolbox *QuTiP* [24], we simulate this process for $\tau = 100$ time steps and obtain the results shown in Fig. 2.

The panel at the top visualizes the behavior of the amplitudes $|a_i(t)|^2$ of the basis states the system could be found in upon measurement. At $t = 0$, all these basis states are equally likely but over time their amplitudes begin to diverge; amplitudes of basis states that correspond to low energy states of our objective increase while amplitudes of basis states that could hardly be considered a solution to our problem decrease.

At $t = \tau$, certain basis states are therefore more likely to be measured than others and the table at the bottom of Fig. 2 ranks the 25 most likely final basis states. Indeed, each of the top 24 most likely states encodes a solution to the 4 rooks problem; the next most likely state does not encode such a solution.

V. Conclusion

Our goal with this paper was to point out that if one knows how to use Hopfield networks for problem solving, one basically also knows how use adiabatic quantum computing for problem solving. This is interesting because Hopfield networks are well known to the neurocomputing community and allow for solving a wide range of typically hard discrete optimization problems. Yet, on the other hand, they suffer from certain inadequacies that can be circumvented on an adiabatic quantum computer.

For instance, a Hopfiled network of n neurons is capable of searching an exponentially large set of 2^n bipolar vectors for a configuration that represents a solution to a problem that has been encoded in terms of an energy function. Updating its neurons in an asynchronous manner, we are guaranteed that the network will converge to a stable state, i.e. to a state that minimizes its energy function. However, it is well known that asynchronous updates realize nothing but a local search process which might get stuck in local minima [25].

An adiabatic quantum computer, on the other hand, performs a global or exhaustive search. This is due to the quantum mechanical phenomenon of superposition which causes all 2^n potential solutions to be considered simultaneously. This, in turn, makes it unlikely that a suboptimal solution will be found. Moreover, whereas a classical exhaustive search would require $O(2^n)$ operations, an adiabatic quantum computer typically requires efforts of only $\sqrt{O(2^n)}$. While this is still exponential in the problem size, the speedup is substantial (see Fig. 3).

The k-rooks problem we considered as a practical example for our discussion was chosen for mere didactic reasons. It does not constitute a difficult problem and, in practice, one would neither solve it using Hopfield networks nor adiabatic quantum computing. However, it exposes how to (re)formulate discrete optimization problems such that they can be solved using Hopfield networks or adiabatic quantum computing. This is interesting, because many much more demanding (i.e. NP complete) optimization problems can be expressed within this energy minimization framework, too [26]. A take home message therefore is that, at the dawn of the era of quantum computing, it might well be worthwhile to reconsider Hopfield networks.

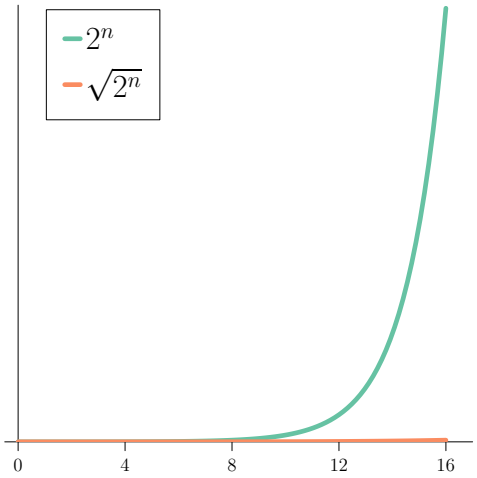


Fig. 3: Illustration of the quadratic speedup achievable via adiabatic quantum computing, i.e. runtimes of $O(\sqrt{2^n})$ rather than of $O(2^n)$.

We also note again that adiabatic quantum computing (as considered in this paper) and quantum gate computing are polynomially equivalent [15]. In other words, if it is possible to express the solution to a problem in terms of minima of a Hamiltonian or energy function such that the problem can be solved on an adiabatic quantum computer (such as a D-Wave device), then the problem can also be solved on a quantum gate computer (for instance an IBM Q device). This can, for example, be achieved using the Suzuki-Trotter expansion [27], [28] or related quantum approximate optimization algorithm techniques (QAOA) [29].

Finally, we should point out that, as of this writing, quantum computing is still "not quite there" yet. In this paper, we saw how quantum mechanical phenomena such as superposition and entanglement are a boon for quantum information processing. Yet, other quantum mechanical phenomena such as decoherence are still its bane. In order for quantum computers to reliably reach quantum supremacy, qubits must be prevented from interacting with the outside world because any such interaction would cause a qubit to collapse to a basis state and therefore eliminate the benefits of superposition. And while the engineering problem of avoiding decoherence becomes more daunting the more qubits need to be manipulated, there is progress in this regard. For instance, in 2018, Google

presented a 72 qubit quantum gate computer and D-Wave began marketing a 2048 qubit adiabatic quantum computer and, in 2019, IBM began selling a 20 qubit quantum gate computer.

It is also important to note that quantum computing should be seen as a form of probabilistic computing. Once a qubit system is measured it collapses to one of its basis states and henceforth behaves just as a system of classical bits. In this paper, we discussed that this collapse happens probabilistically; while appropriate solutions are more likely to be observed, they are not guaranteed to be observed. Any quantum computing algorithm would therefore have to be run several times so as to be sure about the solution obtained. This, however, is a phenomenon not unknown to machine learning practitioners. For example, most algorithms for simple k-means clustering start from random initializations and may thus yield different results in different runs; experts know that several runs are required before conclusions can be drawn.

Despite of these caveats, the potential benefits of quantum computing for unsupervised learning are obvious. In particular, since many problems in this area are subset selection problems in disguise. It should, for instance, be possible to devise Hamiltonians for tasks such as feature selection, similarity searches, k-medoids clustering, or support vector identification.

Acknowledgments

This work is a joint effort of the Fraunhofer Research Center for Machine Learning (RCML) within the Fraunhofer Cluster of Excellence Cognitive Internet Technologies (CCIT) and the Competence Center for Machine Learning Rhine-Ruhr (ML2R) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01IS18038A). The authors gratefully acknowledges this support.

References

[1] M. Cusumano, "The Business of Quantum Computing," *Comm. of the ACM*, vol. 61, no. 10, 2018.

[2] E. Gibney, "Hello Quantum World! Google Publishes Landmark Quantum Supremacy Claim," *Nature*, vol. 574, no. 7779, 2019.

[3] S. Castellanos, "IBM's Quantum Computing Service Top 100 Customers," The Wall Street Journal, Jan. 2020.

[4] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum Clustering Algorithms," in *Proc. ICML*, 2007.

[5] C. Bauckhage, E. Brito, K. Cvejoski, C. Ojeda, R. Sifa, and S. Wrobel, "Ising Models for Binary Clustering via Adiabatic Quantum Computing," in *Proc. EMMCVPR*, 2017.

[6] C. Bauckhage, R. Sifa, and S. Wrobel, "Adiabatic Quantum Computing for Max-Sum Diversification," in *Proc. SIAM SDM*, 2020.

[7] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum Machine Learning," *Nature*, vol. 549, no. 7671, 2017.

[8] V. Dunjiko, J. Taylor, and H. Briegel, "Quantum-Enhanced Machine Learning," *Physical Review Letters*, vol. 117, no. 13, 2016.

[9] M. Schuld, I. Sinayskiy, and F. Petruccione, "An Introduction to Quantum Machine Learning," *Contemporary Physics,*, vol. 56, no. 2, 2014.

[10] N. Wiebe, A. Kapoor, and K. Svore, "Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning," *Quantum Information & Computation*, vol. 15, no. 3–4, 2015.

[11] J. Hopfield, "Neural Networks and Physical Systems with Collective Computational Abilities," *PNAS*, vol. 79, no. 8, 1982.

[12] R. Rojas, *Neural Networks – A Systeatic Introduction*. Springer, 1996.

[13] Z. Bian, F. Chudak, W. Macready, and G. Rose, "The Ising Model: Teaching an Old Problem New Tricks," D-Wave Systems, Tech. Rep., 2010.

[14] M. Johnson and et al., "Quantum Annealing with Manufactured Spins," *Nature*, vol. 473, no. 7346, 2011.

[15] D. Aharanov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation," *SIAM Review*, vol. 50, no. 4, 2008.

[16] P. Shor, "Polynomial-time Algorithms for Prime Factorization and Discrete Logarithm Problems," *SIAM J. on Computing*, vol. 26, no. 5, 1997.

[17] L. Grover, "From Schrödinger's Equation to the Quantum Search Algorithm," *J. of Physics*, vol. 56, no. 2, 2001.

[18] M. Born and V. Fock, "Beweis des Adiabatensatzes," *Zeitschrift für Physik*, vol. 51, no. 3–4, 1928.

[19] T. Albash and D. Lidar, "Adiabatic Quantum Computation," *Reviews of Modern Physics*, vol. 90, no. 1, 2018.

[20] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, 1925.

[21] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum Computation by Adiabatic Evolution," *arXiv:quant-ph/0001106*, 2000.

[22] M. Amin, "Effect of Local Minima on Adiabatic Quantum Optimization," *Physical Review Letters*, vol. 100, no. 13, 2008.

[23] J. Roland and N. Cerf, "Quantum Search by Local Adiabatic Evolution," *Physical Review A*, vol. 65, no. 4, 2002.

[24] J. Johansson, P. Nation, and F. Nori, "QuTiP 2: A Python Framework for the Dynamics of Open Quantum Systems," *Computer Physics Communications*, vol. 184, no. 4, 2013.

[25] X. Huang, "A New Kind of Hopfield Networks for Finding Global Optimum," in *Proc. IEEE IJCNN*, 2005.

[26] A. Lucas, "Ising Formulations of Many NP Problems," *Frontiers in Physics*, vol. 2, 2014.

[27] L.-A. Wu, M. Byrd, and D. Lidar, "Polynomial-Time Simulation of Pairuing Models on a Quantum Computer," *Physical Review Letters*, vol. 89, 2002.

[28] Y. Sun, J.-Y. Zhang, M. Byrd, and L. Wu, "Adiabatic Quantum Simulation Using Trotterization," *arXiv:1805.11568 [quant.ph]*, 2018.

[29] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," *arXiv:1411.4028 [quant.ph]*, 2014.