TU Delft

MSc thesis APPLIED MATHEMATICS

Parameter Optimization for Quantum Annealing

Experimental research on the effects of the Lagrangian multiplier, annealing schedule and the embedding on the performance of standard and reverse quantum annealing

Mirte van Loenen

Delft University of Technology

Supervisor

Dr. rer. nat. Matthias Möller

Additional committee members

Prof. dr. ir. Kees Vuik

Dr. ir. Elena Pulvirenti

July 2nd, 2025

Delft, the Netherlands

Abstract

Quantum annealing is the continuous transformation of the energy working on a quantum system and then measuring said system. The adiabatic theorem and the Ising model let us leverage quantum annealing to find minimal solutions to quadratic unconstrained binary optimization (QUBO) models by setting the energy present during quantum annealing. In this thesis we investigate the QUBO-dization of the multiple object tracking (MOT) problem and perform parameter optimization on the Lagrangian multiplier, chain strength and annealing time parameters as well as test the effects on quantum annealing performance acquired by adding a pause or quench to the anneal schedule or by performing reverse annealing. Experiments were performed on state of the art quantum annealers developed by D-Wave and MOT problem instances were made by us to provide minimally preprocessed QUBO matrices. Reverse annealing was found to have a better performance than standard anneal schedules and we perceived a relation between the annealing time and quantum annealing performance. During testing we found additional factors that contribute to the performance such as the embedding of the Ising model onto the qubits of the quantum annealer. We combine our findings to provide a full quantum annealing schedule and initial state suitable for quantum annealing on state of the art quantum annealers developed by D-Wave.

Contents

0001

Introduction

Quantum annealing is the continuous transformation of the energy working on a quantum system, which is a set of qubits, and then measuring the quantum system. The energy working on a quantum system is described by Hamiltonian operator $\hat{H}(t)$, where time t goes from 0 to end time T. According to the adiabatic theorem, if $\hat{H}(t)$ and the quantum system at the time of initialization have certain properties (given in Section 2.1) and if T is large enough, then we know the relative energy of the quantum state after measurement. This can for example be used to find states with minimal energy, which in turn can be used to calculate the minimal objective function value of quadratic unconstrained binary optimization (QUBO) problems.

In this thesis we will investigate the QUBO-dization of the multiple object tracking (MOT) problem, which is used to assign detections of objects taken from frames to the underlying real-world objects. Examples of MOT include detection of people on camera footage [1] and detection of vehicles on radar measurements [2]. MOT problems and other QUBO problems are often NP-hard, meaning it is expected that there are no polynomial-time algorithms that solve these QUBO and MOT problems. This leads to such increasing amounts of runtime when solving MOT and QUBO problems on many variables, that the optimal results are deemed too computationally expensive to find. Quantum annealing provides a way to solve QUBO problems that differs from classical methods, and may offer improved scalability in the future. With the advances in quantum mechanics and materials science of the past two decades, it is possible to perform quantum annealing on real-world qubits and different instances of QUBO problems have already been solved on quantum annealing hardware [3, 4, 5], including MOT problems [6].

Performing quantum annealing on real-world quantum annealers, or quantum processing units (QPUs), introduces a new set of parameters that are not used in classical methods such as simulated annealing and are difficult to research from a theoretical point of view. Such parameters include the annealing time T, the annealing schedule (explained in Section 6.5) and the chain strength cs, which is used to compensate for missing couplings between qubits within QPUs. Additionally, as MOT problems are not typically unconstrained, we use a Langrangian multiplier approach to implement constraints in a weak sense with the help of a penalty parameter incorporated into the objective function. By combining simulated annealing and quantum annealing experiments, both used to find optimal solutions of our own MOT problem dataset (introduced in Section 3.3), we aim to answer the following questions:

1. How do the annealing time T and the chain strength cs affect the quantum annealing measurement outcomes of the MOT problems considered in this thesis?
2. How does the class of penalty parameters, found using simulated annealing, affect the quantum annealing measurement outcomes of the MOT problems considered in this thesis?
3. Do quantum annealing measurement outcomes obtained from state of the art quantum annealers equal the optimal solution of the MOT problem instances considered in this thesis?

To perform the quantum annealing experiments necessary to answer the research questions, we used the

state of the art quantum annealers from D-Wave. Special thanks to TNO for the access to TNO's Quantum Application Lab Facility, which includes access to the quantum annealers from D-Wave.

Section 2 serves as a short introduction of what quantum annealing is in theory, including how it is used to solve optimization problems. In Section 3 an introduction to multiple object tracking and its formal problem description are given. In Section 3.3 we define a simple MOT dataset that will be used as input for all annealing experiments. Starting from Section 4, the aspects of quantum annealing related to real-world QPUs are discussed. In Section 5 we use quantum annealers to create and discuss some preliminary results using our earlier defined dataset. The performance of the preliminary experiments will set the bar which we will try to improve upon, mostly by parameter engineering, in Section 6. Finally, we provide conclusions in Section 7 and we some context, discussion and suggestions for future work in Section 8.

Acknowledgements

We would like to again thank TNO for providing access to D-Waves's quantum annealers. Thank you Roos van Loenen for providing MOT instance frames on such a short notice, even though they were eventually not used. Also, thank you to Birgit van Prooijen, Zylan Benjert, and especially Quinten Donker for proofreading. Extra thanks to Birgit van Prooijen for helping with the reverse annealing figures and Quinten Donker for helping with the detection visualisations and thank you both for sitting next to me during the project, as working while sitting between the two of you is my favourite spot. Thank you to Waded Oudhuis, Rick Schermerhorn, Julius Strack van Schijndel, and my parents for pushing me to start this thesis and all the support. Thank you to Kees Vuik and Elena Pulvirenti for joining the thesis committee. Finally, special thanks to Matthias Möller for making this thesis possible, supervising the entire project, no matter how busy, and sharing the high level of enthusiasm on the topic of quantum annealing.

0010

Quantum Annealing

In one sentence, quantum annealing is a computational method that slowly changes the energy working on a quantum system and then measures said system. To further elaborate on this, we first give some definitions and notation of the energy evolution of a quantum system. How the energy of such a system changes is explained in Section 2.1 and how this relates to optimization problems is explained in Section 2.2. For an introduction to quantum mechanics we refer to "Quantum Computation and Quantum Information" by Nielsen and Chuang [7].

Hamiltonians are the operators corresponding to the energy of a system and are described by Hermitian matrices, that is matrices $H \in \mathbb{C}^{n\times n}$ that satisfy the condition $H^\dagger = H$. Recall that Hermitian matrices are diagonalizable with real eigenvalues. The evolution over time of the state of a quantum system $|\phi(t)\rangle$ is described by the time-dependent **Schrödinger equation**:

$$i\hbar\frac{\partial}{\partial t}|\phi(t)\rangle = \hat{H}(t)|\phi(t)\rangle, \tag{2.1}$$

where $\hbar$ is the reduced Planck constant and Hamiltonian $\hat{H}(t)$ is the time-dependent operator working on the system. The eigenvalues E_k, which are all real, of the Hamiltonian are possible energy levels within the system and its eigenstates, also known as eigenvectors, are the corresponding quantum states $|\phi_k\rangle$. For convenience the eigenvalues are ordered in ascending order, thus $E_0 \leq E_1 \leq E_2 \ldots \leq E_n$ are used to denote the eigenvalues of the Hamiltonian acting on the system. In fields such as quantum annealing, it is useful to look at the lowest possible energy of the system, which is why it is separately defined.

Definition 2.1. The **ground state energy** of a Hamiltonian $\hat{H}(t)$ at time t is its lowest eigenvalue E_0 [8], i.e. the minimal value

$$E_0 := \min_{|\phi\rangle}\langle\phi|\hat{H}(t)|\phi\rangle.$$

Of course, it is also interesting to know for which state the system's energy reaches the ground state energy.

Definition 2.2. The **ground state** of a Hamiltonian $\hat{H}(t)$ at time t is its eigenstate $|\phi_0\rangle$ corresponding to eigenvalue E_0 [8], i.e. the state

$$|\phi_0\rangle := \arg\min_{|\phi\rangle}\langle\phi|\hat{H}(t)|\phi\rangle.$$

In quantum annealing the Hamiltonian $\hat{H}(t)$ is dependent on time. Therefore, the key point of interest in quantum annealing is what happens when blending one Hamiltonian into another in a continuous fashion.

2.1. Adiabatic evolution

Quantum annealing is the adiabatic evolution of the energy working on the quantum state. Adiabatic, according to Oxford dictionary, is "relating to or denoting a process or condition in which heat does not enter

or leave the system concerned", i.e. a closed system, which is currently more of an ideal than a reality for quantum systems. Before discussing the differences between annealing in theory and in practice (which is done in Section 4), we first give the theory.

In the quantum annealing process the system starts in the initial state $|\phi(t=0)\rangle$ with initial Hamiltonian $\hat{H}_I := \hat{H}(t=0)$ and the process stops at time $t=T$ with problem Hamiltonian $\hat{H}_P := \hat{H}(t=T)$. Note that this is the general understanding of quantum annealing, but does not have to be the case. For example, for reverse annealing, which is a type of quantum annealing that can be performed on D-Wave quantum annealers and will be explained in Section 4.1, the initial state and Hamiltonian behave in a different manner.

Not only is the ground state energy of interest, but also the difference between the ground state energy E_0 and the second to lowest energy E_1, which is called the energy of the first excited state.

Definition 2.3. A Hamiltonian $\hat{H}$ is **gapped** for t with $0 \leq t \leq T$, if

$$\delta(t) := E_1(t) - E_0(t)$$

is strictly positive [9].

Definition 2.4. The **minimal energy gap** of Hamiltonian $\hat{H}(t)$ is equal to

$$\delta_{\min} := \min_{t\in[0,T]} \delta(t) \geq 0.$$

If $\delta_{\min} > 0$, it is said that $\hat{H}$ is gapped for all t [9].

Using the definitions above, quantum annealing is based on the following version of the adiabatic theorem.

Theorem 2.1. ***The adiabatic theorem (simplified version of [10])*** *Let $\hat{H}(t)$ be a smooth family of Hamiltonians parametrized by $t \in [0,T]$, with $\hat{H}_I = \hat{H}(t=0)$, and $\hat{H}_P = \hat{H}(t=T)$, such that $\delta_{\min} > 0$. If the system starts in the ground state of the initial Hamiltonian, $|\phi(t=0)\rangle = |\phi_0\rangle_{\hat{H}_I}$, T is large enough, and the system evolves following the time-dependent Schrödinger equation (2.1) with $\hat{H}(t)$, then the system ends in the ground state of the final Hamiltonian, i.e. state $|\phi(T)\rangle = |\phi_0\rangle_{\hat{H}_P}$.*

There are different versions of the adiabatic theorem. For example, in 1958, Albert Messiah [11] wrote that $|\phi(T)\rangle$ has the asymptotic property under (roughly) the same conditions as in Theorem 2.1, which is then used to give the adiabatic approximation of system $|\phi(T)\rangle$ and then to derive the same results as in Theorem 2.1. This result, however, can be used to approximate the final system $|\phi(T)\rangle$ for every $|\phi(0)\rangle$. The proof of this version is based on that, because the eigenvalues (of interest) do not cross, the final state of the system becomes a function of only the solution of the Schrödinger equation and the operators that project onto the eigenspaces of $\hat{H}$ when T goes to infinity. The actual proof including the asymptotic property and the adiabatic approximation are beyond the scope of this report. The proof of the adiabatic theorem (similar to Theorem 2.1) provided by Born and Fock in [10] from 1928 is easier to understand, as Born and Fock directly show that the probability of going from the desired state, i.e. the ground state of $\hat{H}(t)$ for $t \in [0,T]$, to any other state goes to zero as T goes to infinity.

Both proofs heavily rely on $\delta_{\min} > 0$, but in 1999 Avron and Elgart gave a proof of the adiabatic theorem without this gap condition [12]. This and the previously mentioned proofs are not used in the remainder of this report, as the aim of the report is rather to show the practical side of quantum annealing instead of the theoretical side. We do still recommend reading the proofs to readers interested in quantum mechanics on edge cases such as T going to infinity.

The adiabatic theorem does not only hold for the ground state, but for any state $|\phi_i\rangle$ corresponding to E_i, as long as $|\phi_{i-1}\rangle < |\phi_i\rangle < |\phi_{i+1}\rangle$ holds for all $0 \leq t \leq T$ [10]. However, because the ground state can be related to the minimum value of an optimization problem, we are only interested in the ground state and the first excited state.

The adiabatic theorem is used for certain optimization problems, where the initial Hamiltonian and its ground state are known and the ground state of the problem Hamiltonian is unknown and difficult to compute using other techniques. Let us now discuss the connection between quantum annealing and optimization problems.

2.2. Optimization Problems for Quantum Annealing

First, we discuss an example of annealing from Farhi [9]. Then we take a small step to annealing in practice by looking at Hamiltonians that are of Ising model form and rewriting them as optimization problems.

Example of a quantum annealing problem

In 2000 Farhi et al. published a paper called "Quantum Computation by Adiabatic Evolution" [9], which contains multiple interesting problems and their construction of the corresponding initial and problem Hamiltonians as well as the eigenvalues of $\hat{H}(t)$. The examples provided by Farhi are an excellent introduction to constructing the (initial and) problem Hamiltonian, which we will also do in Section 3. We will now discuss Farhi's "Three Qubits" example in detail.

To construct a problem Hamiltonian, Farhi introduces some requirements one can impose on (qu)bits. The three requirements that Farhi uses are sets of two qubits that agree, disagree or a set of two qubits where one implies the other, which is when one qubit is in a state greater than or equal to the state of the other qubit. Qubit i and j agree with each other if they are in the same state, otherwise they disagree. Qubit i implies j holds for states $|ij\rangle \in \{|00\rangle, |01\rangle, |11\rangle\}$. See Table 2.1 for the truth table.

Table 2.1: Truth table of states of qubits i and j and the constraints "implies", "agree" and "disagree".

Qubit i	Qubit j	i implies j	i and j agree	i and j disagree
0	0	1	1	0
0	1	1	0	1
1	0	0	0	1
1	1	1	1	0

To demonstrate a quantum annealing process we choose the following problem Hamiltonian by Farhi [9]:

Consider a 3-qubit system, where the desired state is qubit 1 implies 2, qubit 1 and 3 disagree and qubit 2 and 3 agree. Agree, disagree and imply can be written as matrix operators, where the lowest eigenvalues correspond to the desired states.

To construct the corresponding problem Hamiltonian, we view the three requirements separately, construct their problem Hamiltonians, and then construct the final problem Hamiltonian by taking the sum. The problem Hamiltonians for "agree", "disagree", and "implies" all give the values from Table 2.1 and are constructed with the identity and Pauli operators in the following way [9]:

$$\hat{H}^{(ij)}_{\text{agree}} = \frac{1}{2}\left(\mathbb{1}^{(ij)} - \hat{\sigma}^{(i)}_z \otimes \hat{\sigma}^{(j)}_z\right), \tag{2.2}$$

$$\hat{H}^{(ij)}_{\text{disagree}} = \mathbb{1}^{(ij)} - \hat{H}^{(ij)}_{\text{agree}} = \mathbb{1}^{(ij)} - \frac{1}{2}\left(\mathbb{1}^{(ij)} - \hat{\sigma}^{(i)}_z \otimes \hat{\sigma}^{(j)}_z\right) = \frac{1}{2}\left(\mathbb{1}^{(ij)} + \hat{\sigma}^{(i)}_z \otimes \hat{\sigma}^{(j)}_z\right), \tag{2.3}$$

$$\hat{H}^{(ij)}_{\text{imply}} = \frac{1}{2}\left(\mathbb{1}^{(i)} - \hat{\sigma}^{(i)}_z\right) \otimes \frac{1}{2}\left(\mathbb{1}^{(j)} + \hat{\sigma}^{(j)}_z\right). \tag{2.4}$$

The problem Hamiltonian then becomes the sum of its three parts, $\hat{H}_P = \hat{H}^{(12)}_{\text{imply}} + \hat{H}^{(13)}_{\text{disagree}} + \hat{H}^{(23)}_{\text{agree}}$.

The initial Hamiltonian is constructed in a similar fashion. If the problem Hamiltonian is a sum of $M \in \mathbb{Z}_{>0}$ Hamiltonians ($M = 3$ in the case of this "Three Qubits" example), and for each $m \in \{1, \ldots, M\}$ Hamiltonian $\hat{H}_m$ works on a subset S_m of the qubits, then the initial Hamiltonian becomes the sum of $\sum_m |S_m|$ Hamiltonians. Specifically, for $\hat{H}_P = \sum_m \hat{H}_m^{(S_m)}$ Farhi uses Equation 2.5 to construct the initial Hamiltonian [9]:

$$\hat{H}_I = \sum_m \sum_{i \in S_m} \hat{H}^{(i)}_B, \quad \hat{H}^{(i)}_B = \frac{1}{2}\left(\mathbb{1}^{(i)} - \sigma^{(i)}_x\right). \tag{2.5}$$

For this example the initial Hamiltonian then becomes $\hat{H}_I = (\mathbb{1}^{(1)} - \sigma^{(1)}_x) + (\mathbb{1}^{(2)} - \sigma^{(2)}_x) + (\mathbb{1}^{(3)} - \sigma^{(3)}_x) = 3\mathbb{1}_8 - \sigma_x \otimes \mathbb{1} \otimes \mathbb{1} - \mathbb{1} \otimes \sigma_x \otimes \mathbb{1} - \mathbb{1} \otimes \mathbb{1} \otimes \sigma_x$. Farhi deems this a suitable initial Hamiltonian as "since the Hilbert space can be

decomposed into a direct sum of the invariant and odd subspaces and accordingly $\hat{H}(t)$ is block diagonal, the invariant and odd states are decoupled, and their crossing is not an unlikely occurrence" [9]. This will also be the case for the initial state used on D-Wave quantum annealers, discussed in Section 4.1.

Now that the initial and problem Hamiltonians are defined, we need to construct the time evolution of the Hamiltonian working on the system, $\hat{H}(t)$. This is typically (see, e.g. [9, 6]) described as

$$\hat{H}(t) = \frac{T-t}{T}\hat{H}_I + \frac{t}{T}\hat{H}_P. \tag{2.6}$$

Because T is not the same for every annealing process, it is easier to consider anneal fraction $s := \frac{t}{T}$. Therefore, Equation 2.6 is the same as

$$\hat{H}(s) = (1-s)\hat{H}_I + s\hat{H}_P. \tag{2.7}$$

From the now fully defined $\hat{H}(s)$ we want to know the eigenvalues. These are displayed in Figure 2.1 for all s from 0 to 1. $\hat{H}$ is indeed gapped for all t, as it is gapped for all s, and the minimal gap is $\delta_{\min} \approx 0.5$ and is reached at anneal fraction $s \approx 0.72$.

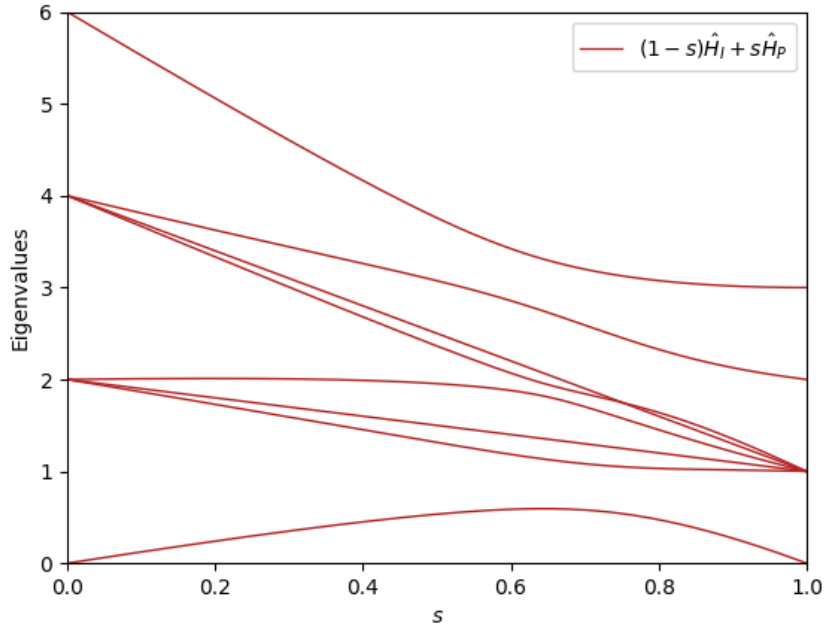

Figure 2.1: Eigenvalues of $\hat{H}(s)$ as defined by Farhi [9].

Quadratic Unconstrained Binary Optimization problems

In practice, we exclusively look at Hamiltonians that can be written as an Ising model:

$$\hat{H}_{\text{Ising}} = \sum_{i<j} \mathbf{J}_{i,j}\boldsymbol{\sigma}_i\boldsymbol{\sigma}_j + \sum_i \mathbf{h}_i\boldsymbol{\sigma}_i, \tag{2.8}$$

where $\boldsymbol{\sigma}_i \in \{-1, 1\}$ is the spin of particle i and elements of matrix $\mathbf{J}$ and vector $\mathbf{h}$ can be set on a quantum annealing system. The physical properties that make up $\mathbf{J}$ and $\mathbf{h}$ are discussed in Section 4. Because $\mathbf{J}$ and $\mathbf{h}$ can be set on a quantum annealer, it is necessary to write the initial Hamiltonian and the problem Hamiltonian in Ising form. The ground state of $\hat{H}_{\text{Ising}}$ is the state

$$\underset{\boldsymbol{\sigma}\in\{-1,1\}^n}{\arg\min} \sum_{i<j} \mathbf{J}_{i,j}\boldsymbol{\sigma}_i\boldsymbol{\sigma}_j + \sum_i \mathbf{h}_i\boldsymbol{\sigma}_i = \underset{\boldsymbol{\sigma}\in\{-1,1\}^n}{\arg\min} \frac{1}{2}\boldsymbol{\sigma}^T\mathbf{J}\boldsymbol{\sigma} + \mathbf{h}^T\boldsymbol{\sigma}, \tag{2.9}$$

where $\mathbf{J}$ is assumed to be symmetric with zeroes on the diagonal. To relate the spin of a particle to a decision problem, we introduce decision variable $\mathbf{z}_i \in \{0,1\}$ for every $\boldsymbol{\sigma}_i \in \{-1,1\}$, by defining $\mathbf{z} := \frac{1}{2}(\boldsymbol{\sigma} + \mathbf{1})$. Rewriting Equation 2.9 in terms of $\mathbf{z}$ then gives the following decision problem

$$\begin{aligned}
\underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, \frac{1}{2}(2\mathbf{z}-\mathbf{1})^T\mathbf{J}(2\mathbf{z}-\mathbf{1}) + \mathbf{h}^T(2\mathbf{z}-\mathbf{1}) &= \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, 2\mathbf{z}^T\mathbf{J}\mathbf{z} - \mathbf{1}^T\mathbf{J}\mathbf{z} - \mathbf{z}^T\mathbf{J}\mathbf{1} + \frac{1}{2}\mathbf{1}^T\mathbf{J}\mathbf{1} + 2\mathbf{h}^T\mathbf{z} - \mathbf{h}^T\mathbf{1} \\
&= \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, 2\mathbf{z}^T\mathbf{J}\mathbf{z} - 2\mathbf{1}^T\mathbf{J}\mathbf{z} + 2\mathbf{h}^T\mathbf{z} + \frac{1}{2}\mathbf{1}^T\mathbf{J}\mathbf{1} - \mathbf{h}^T\mathbf{1} \\
&= \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, 2\mathbf{z}^T\mathbf{J}\mathbf{z} + (-2\mathbf{J}^T\mathbf{1} + 2\mathbf{h})^T\mathbf{z} + \frac{1}{2}\mathbf{1}^T\mathbf{J}\mathbf{1} - \mathbf{h}^T\mathbf{1} \\
&\overset{(*)}{=} \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, 2\mathbf{z}^T\mathbf{J}\mathbf{z} + (-2\mathbf{J}^T\mathbf{1} + 2\mathbf{h})^T\mathbf{z} \\
&\overset{(**)}{=} \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, \mathbf{z}^T\mathbf{J}\mathbf{z} + (-\mathbf{J}^T\mathbf{1} + \mathbf{h})^T\mathbf{z} \\
&= \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, \mathbf{z}^T\mathbf{J}\mathbf{z} + \mathbf{h}'^T\mathbf{z} \\
&\overset{(***)}{=} \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\, \mathbf{z}^T\mathbf{J}'\mathbf{z},
\end{aligned}$$

where step $(*)$ removes the terms that do not rely on $\mathbf{z}$ and therefore do not influence the optimal $\mathbf{z}$, $(**)$ scales the entire objective function without influencing optimal $\mathbf{z}$, and $(***)$ has the elements of $\mathbf{h}' = -\mathbf{J}^T\mathbf{1} + \mathbf{h}$ on the diagonal of $\mathbf{J}'$. Step $(***)$ is only possible because $\mathbf{z}$ consists of elements such that $\mathbf{z}_i^2 = \mathbf{z}_i$ for every i.

Note that the equation above is of the form of a quadratic unconstrained binary optimization (QUBO) problem. Because of the invertible operations performed, this means it is possible to solve any QUBO problem on a quantum annealer by rewriting the QUBO form as an Ising model, using that as the problem Hamiltonian and finding its ground state using a quantum annealer.

There is an abundance of NP-hard problems that can be formulated as QUBO problems [13] and in Ising model form [14]. Because these problems are NP-hard they are difficult to solve when scaling up the amount of variables. Quantum annealing may provide the necessary speed-up to solve large instances of these problems in achievable amounts of time. To keep the focus of this report on the performance of quantum annealing, we will focus on one type of QUBO problem: multiple object tracking.

0011

Problem Description

Multiple Object Tracking (MOT) is an example of an assignment problem where detections of possible objects are assigned to true objects, called tracks. If detections are made at different times, for example if a measurement is made every second, then each detection/measurement instance is called a frame and every frame has its own set of detections. The tracks are set and not dependent on the frames, as they represent real objects and exist independently of how they are measured [1]. The goal is to assign the detections to the tracks in a way that maximizes the similarities between detections and tracks as well as the similarities between detections from different frames that are assigned to the same track.

There are multiple ways of describing a MOT problem as an optimization problem. We start by describing the problem using the constraints and parts of the notation from [6]. Then in Section 3.2 the problem will be restated as a QUBO problem using Lagrangian optimization. To tailor these problem descriptions to quantum annealing, we wish to use simple datasets. Therefore, some datasets are generated using a process detailed in Section 3.3.

3.1. Multiple Object Tracking

In mathematical terms, a multiple object tracking problem is described as follows: For every frame $f \in \{1, \ldots, F\} = [F]$ there are D_f detections and one dummy detection. Each detection d_f is assigned to a fixed set of tracks $t \in [T+1]$, where track $t = T+1$ represents the dummy track instead of the track of an existing object. To store the assignments of detections in frame f, we use assignment matrix $\mathbf{X}_f \in \{0,1\}^{(D_f+1)\times(T+1)}$. If in frame f detection d_f is assigned to track t, decision variable $x_{f d_f t} = (\mathbf{X}_f)_{d_f t} = 1$, and 0 otherwise. The total number of decision variables is $n = \sum_{f=1}^{F}(D_f+1)(T+1) = (T+1)\left(F + \sum_{f=1}^{F} D_f\right)$.

Assigning detections to tracks comes with its own set of rules. First, a detection has to be assigned to a track, as a detection represents the measurement or simplified version of one object so it must be assigned to a track representing a true object. Additionally, because different tracks represent different distinct objects, assigning a detection to multiple tracks would mean that a single detection simultaneously belongs to different objects. Therefore, a detection must be assigned to exactly one track. Following this logic, a track can not have multiple detections from the same frame assigned to it, as that would mean the track, representing a singular object, is detected as multiple objects. Furthermore, it is required that a track is detected, meaning that there must be a detection that is assigned to that track at every frame or measuring instance. Therefore, for every frame, a track must have exactly one detection assigned to it.

[1] This means MOT is not suitable for situations where objects do not exist independently of the measurements, for example during a double slit experiment.

(a) Frame 1 with 1 detection of a Pallas's cat.

(b) Frame 2 with 1 detection of a Pallas's cat.

(c) Frame 3 with 0 detections of Pallas's cats.

(d) Frame 4 with 1 detection of a Pallas's cat.

Figure 3.1: Example of 4 consecutive frames with the Pallas's cat being 1 track. The track moves between the frames and is not detected in frame 3 as the track is behind the rock. As a result, the dummy detection variable must be assigned to track 1 (out of 1) to ensure the constraints imposed on detections in frame 3 are met.

To ensure detections are assigned exactly once and exactly one detection per frame is assigned to a track, the following one-hot encodings act as constraints:

$$\sum_{d_f=1}^{D_f+1} x_{fd_ft} = 1, \forall f \in [F], \forall t \in [T], \tag{3.1}$$

$$\sum_{t=1}^{T+1} x_{fd_ft} = 1, \forall f \in [F], \forall d_f \in [D_f]. \tag{3.2}$$

Note that these constraints are not enforced for dummy detections $d_f = D_f + 1$, $f \in [F]$ and dummy track $t = T+1$, as they do not represent objects. In total there are m number of constraints, with $m = FT + \sum_{f=1}^{F} D_f = \sum_{f=1}^{F}(D_f + T)$. If all tracks are detected in each frame exactly once, the dummy variables serve no purpose and remain equal to zero. At least one of the dummy variables is unequal to zero in the following situations:

- If in a frame f there are more detections than tracks, so $D_f > T$, there is at least one detection remaining after assigning each track a detection. The remaining detections are assigned to the dummy track.
- If in a frame f not all tracks are detected, $D_f < T$, there will be at least one track that is not given an assigned detection. In that case the dummy detection of that frame is assigned to all such tracks.

Maximizing the similarities of detections across different frames assigned to the same track is the same as minimizing the cost of assigning detections to the same track. The cost between two frames f and g is then the sum of all costs of pairs of detections that are assigned to the same track.

$$c_{fg} = \sum_{t} \sum_{d_f} \sum_{d_g} x_{fd_ft} q_{d_fd_g} x_{gd_gt}, \tag{3.3}$$

where $q_{d_fd_g}$ is the *same assignment cost* (called similarity score in [6]) between detection d_f from frame f and detection d_g from frame g. The values of $q_{d_fd_g}$ can be set in many different ways and will be further discussed in Section 3.3. Note that the track does not influence the value of $q_{d_fd_g}$. Therefore, $q_{d_fd_g}$ is used instead of $q_{d_ftd_gt}$ as it would be the same for all $t \in [T+1]$.

As done in [6], we can set $\mathbf{z}_f := \text{vec}(\mathbf{X}_f)$, with vec($\mathbf{X}$) as a row-major vectorization.

Definition 3.1. A row-major vectorization of an $n \times m$ matrix $\mathbf{X}$ is the vector
$\text{vec}(\mathbf{X}) := [\text{row } 1(\mathbf{X}) \text{ row } 2(\mathbf{X}) \ldots \text{row } n(\mathbf{X})]^T$.

We can then rewrite Equation 3.3 as a quadratic form

$$c_{fg} = \mathbf{z}_f^T \mathbf{Q}_{fg} \mathbf{z}_g, \tag{3.4}$$

with $\mathbf{Q}_{fg}$ the similarity score matrix of size $(D_f+1)\times(D_g+1)$ between frames f and g. By further concatenating we obtain $\mathbf{z} := [\mathbf{z}_1 \dots \mathbf{z}_F]^T$ and block-matrix $\mathbf{Q}$ made out of $\mathbf{Q}_{fg}$ of all frame combinations. The total same assignment cost is then written as

$$c_{sa} = \sum_{f=1}^{F}\sum_{g=1}^{F} c_{fg} = \mathbf{z}^T\mathbf{Q}\mathbf{z}. \tag{3.5}$$

c_{sa} is the total cost of assigning detections to the same track, and independent of which track a set of detections is assigned to. In [6] the cost $\mathbf{b}_{d_f t}$ of assigning a detection d_f to a track t is not discussed, though, it is part of the objective function. The assignment cost is denoted as

$$s_a = \mathbf{b}^T\mathbf{z} = \sum_{f=1}^{F}\mathbf{b}_f^T\mathbf{z}_f = \sum_{f=1}^{F}\mathbf{b}_f^T\mathrm{vec}(\mathbf{X}_f) = \sum_{f=1}^{F}\sum_{d_f=1}^{D_f+1}\sum_{t=1}^{T+1}\mathbf{b}_{d_f t}x_{f d_f t}. \tag{3.6}$$

Adding the same assignment cost from Equation 3.5 and the assignment cost from Equation 3.6, the mathematical description of a MOT problem then becomes the following:

$$\underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\ \mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{b}^T\mathbf{z} \quad \text{s.t. } \mathbf{G}\mathbf{z} = \mathbf{y}, \tag{3.7}$$

where $\mathbf{G}\mathbf{z} = \mathbf{y}$ stores all equality constraints from Equations 3.1 and 3.2.

In order to write the MOT problem from Equation 3.7 as a QUBO, we introduce a penalty term to the objective function for every constraint that is not satisfied. This can be done using Lagrangian optimization.

3.2. Lagrangian Optimization

Lagrangian optimization is a way to eliminate the constraints by adding a penalty term to the objective function. Because we are minimizing we want to add a non-negative penalty term that is smallest (i.e. equal to zero) when all constraint are met. Using a Lagrangian multiplier the unconstrained problem becomes

$$\underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\ \mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{b}^T\mathbf{z} + \lambda\left\|\mathbf{G}\mathbf{z}-\mathbf{y}\right\|_2^2, \tag{3.8}$$

with $\lambda > 0$. Note that because $\lambda\left\|\mathbf{G}\mathbf{z}-\mathbf{y}\right\|_2^2 = \lambda\langle\mathbf{G}\mathbf{z}-\mathbf{y}, \mathbf{G}\mathbf{z}-\mathbf{y}\rangle = \lambda\mathbf{z}^T\mathbf{G}^T\mathbf{G}\mathbf{z} - 2\lambda\mathbf{y}^T\mathbf{G}\mathbf{z} + \lambda\mathbf{y}^T\mathbf{y}$, and $\lambda\mathbf{y}^T\mathbf{y}$ is not dependent on $\mathbf{z}$, solving Equation 3.8 is the same as solving

$$\underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\ \mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{b}^T\mathbf{z} + \lambda\mathbf{z}^T\mathbf{G}^T\mathbf{G}\mathbf{z} - 2\lambda\mathbf{y}^T\mathbf{G}\mathbf{z} = \underset{\mathbf{z}\in\{0,1\}^n}{\arg\min}\ \mathbf{z}^T\mathbf{Q}'\mathbf{z} + \mathbf{b}'^T\mathbf{z}, \tag{3.9}$$

with $\mathbf{Q}' = \mathbf{Q} + \lambda\mathbf{G}^T\mathbf{G}$ and $\mathbf{b}' = \mathbf{b} - 2\lambda\mathbf{G}^T\mathbf{y}$. The higher the value of λ, the higher the penalty becomes, therefore λ must be chosen large enough to ensure that all equality constraints hold. Simultaneously, choosing λ too large decreases the odds of finding the optimal solution, as the energy difference between feasible solutions becomes negligible. Additionally, increasing λ may result in scaling problems on annealing systems.

Note that using Lagrangian optimization to rewrite a quadratic constrained binary optimization problem as a QUBO problem is a method not exclusive to MOT problems and that any quadratic binary optimization problem, constrained or not, can be solved using a quantum annealer, as discussed in Section 2.2.

3.3. Datasets

There are many different MOT problems and as the cost values fully determine the objective function, there are also many different ways of calculating the cost values. There exist benchmark datasets such as the "2D MOT15" dataset [1], which is used by [6]. The downside of this dataset, however, is the cost calculation. The costs, or similarity scores, are calculated using machine learning. This may optimize the values, but makes it more difficult to identify its influence on the results. As we are interested in the performance of quantum

annealing, we made two simple datasets: a “small” and a “large” dataset. Both datasets consist of tracks and detections in one dimension, i.e. lines/dots on one axis. Both datasets also have the same method of detection and cost calculation. We will now discuss all aspects of the datasets in detail.

Tracks and detections

First we discuss going from a set of tracks to a set of detections within one frame. Each track t has four parameters: index (t), left-coordinate (l_t), right-coordinate (r_t), and velocity per frame (v_t). In each frame, the left- and right-coordinates of the tracks are detected in order of index, i.e. if the index of track t is smaller than the index of track s, then track t is detected first and affects the detection of track s, similar to looking at two objects of which one is in front of the other. The velocity is not detected and tracks can only be detected within set bounds (B_l, B_r).

Table 3.1: Cases where the detection parameters of detection d differ from the track parameters of track t. s is any track index lower than index t. Legend for the visuals column: Red is the boundary, blue is track t, shaded blue is part of track t that is not detected, green is track s.

Case	Detection	Interpretation	Visual
$l_t < B_l < r_t$	$l_d = B_l$	Left part of track is out of bounds	
$l_t < B_r < r_t$	$r_d = B_r$	Right part of track is out of bounds	
$r_t < B_l$ or $l_t > B_r$	None	The track is fully out of bounds	
$l_s \leq l_t < r_s < r_t$	$l_d = r_s$	Left part of track is behind track s	
$l_t < l_s < r_t \leq r_s$	$r_d = l_s$	Right part of track is behind track s	
$l_t < l_s < r_s < r_t$	$d = (l_t, l_s)$ and $d+1 = (r_s, r_t)$	Middle part of track is behind track s	
$l_s \leq l_t < r_t \leq r_s$	None	Track is fully behind track s	
$l_t = r_t$	None	Track is too small to detect	

Then, all tracks are detected in ascending index order. Each detection contains an index, a left-coordinate, and a right-coordinate and is stored as $d = (l_d, r_d)$. The first detection that is made has index 1, the second detection has index 2, etc. until there is nothing left that can be detected, resulting in D_f detections. If all tracks are detected perfectly, then for each track t we would have detection d such that index $d = t$, $l_d = l_t$, and $r_d = r_t$. This exact correct detection does not always occur, however, and there are many cases for

which a detection does not equal a track. See Table 3.1 for all cases where the detection does not equal its corresponding track.

A detection is compared to the corresponding (same index) track and updated according to Table 3.1. Then the detection is compared to the track with an index of one lower ($s = t-1$), as that would be directly in front. Then this is done for every track in front, so every track with lower index, until we are left with the final detection parameters of detection d. If no detection is made, the same index d will be used to correspond to the track with one index higher, i.e. empty detections are thrown out and instead of skipping one for the index, the index is reused for the next detection, which corresponds to the next track. This process of detecting a track and comparing to tracks "in front" is done for every track.

For each frame $f > 1$ the parameters of each track are updated: $l_t = l_t + v_l$, $r_t = r_t + v_t$. v_t can also be dependent on the frame, i.e. $v_t(f)$, which is the case for one track in the large dataset at the end of this section.

Costs

Once the detections are made for every frame, we can construct **Q** and **b**, that is, we must set a way to calculate the (same) assignment costs. To use quantum annealing on D-Wave annealers one can either input the QUBO matrix or the Ising model. If the QUBO is used as input, D-Wave automatically transforms it into an Ising model before performing quantum annealing. To construct the costs we chose to give the input in Ising model form, therefore we give **J** (as a symmetric matrix) and **h** (as vector). Because of the similarities between the QUBO problem and Ising model formulations we still refer to the elements of **J** and **h** as (same) assignment costs.

The index is not taken into account, so the cost function may only depend on the left-coordinates, the right-coordinates, and in the case of same assignment costs, the difference in frames. To maximize the similarity, the cost is minimized, i.e. for detections/tracks with similar parameters we want a lower cost value. For our dataset we define the assignment cost of assigning detection d_f to track t as follows:

$$\mathbf{h}_{d_f t} = \begin{cases} 1, & \text{if } r_t < l_{d_f} \text{ or } r_{d_f} < l_t, \\ -1, & \text{otherwise.} \end{cases} \tag{3.10}$$

The same assignment cost of detection d_f from frame f and detection d_g from frame g is set to:

$$\mathbf{J}_{d_f d_g} = \begin{cases} -1, & \text{if } \frac{2}{3} < \frac{r_{d_f} - l_{d_f}}{r_{d_g} - l_{d_g}} < \frac{3}{2}, \\ 0, & \text{if } d_f \text{ or } d_g \text{ is a dummy or } f = g \text{ or } \left|f-g\right| > 4, \\ 1, & \text{otherwise.} \end{cases} \tag{3.11}$$

where $l_t = r_t = 0$ if t denotes the dummy track and $l_{d_f} = r_{d_f} = 0$ denotes the dummy detection. It is easy to see that these costs are not unique and there are many different ways to set the costs, even after normalization. There are two reasons why we chose these costs: to compare with previous work on quantum annealing for MOT problems ([6], where costs are equal to either 0, 1, or -1 as well) and to compare to other Ising models set on quantum annealers, e.g., (random sparse) matrices with elements of 0, 1, or -1 [15, 16, 17].

Note that detections of two frames are not taken into account when the frames considered have over three frames between them, meaning their indices differ by more than four. This can of course be changed, however, keep in mind that the sparsity of **J**, and **Q**, will then also be changed.

Variable reduction

The amount of decision variables is denoted by n. Recall that we currently have $n = (T+1)\left(F + \sum_{f=1}^{F} D_f\right)$. This includes having $T+1+D_f+1-1$ decision variables related to the dummy track and dummy detection of each frame. Here, we will discuss the use of dummy track and dummy detection and eliminate the corresponding variables where possible. Fewer variables equals fewer qubits necessary on a quantum annealer, which is a considerable upside of looking into variable reduction.

Recall the use of the dummy track and dummy detection from Section 3.1: If $D_f < T$, then there are tracks

remaining even if every detection is assigned to a different track. A dummy detection is used to assign to the remaining tracks. If $D_f > T$, then it is not possible to assign all detections to different tracks, so a dummy track is used to make sure the constraints on having exactly one detection assigned to each track are not violated. For any amount of tracks and detections exactly one of the following cases must hold: 1. $D_f < T$, 2. $D_f > T$, or 3. $D_f = T$. In case 1 there is no need for a dummy track, in case 2 there is no need for a dummy detection and in case 3 there is no need for either. Therefore, instead of having $(T+1)(D_f+1)$ decision variables per frame, we have $T(D_f+1)$, $(T+1)D_f$, or TD_f variables if frame f falls under case 1, 2, or 3, respectively. For example, if there are 2 tracks and 3 frames, where in one frame only one detection is made, then originally there are $n = 3(3+5) = 24$ variables, but after removing the unnecessary dummy tracks and detections we have $n = 2(2 \cdot 2)+1(2(1+1)) = 12$, which is only half of the variables! See Figure 3.2 for examples of our dataset.

Dataset specifications

Everything necessary to make a MOT dataset is discussed, except the trajectories of the tracks. We have made two datasets: a small and a large one. The small dataset contains up to 3 tracks and 1 up to 10 frames. The large dataset contains up to 4 tracks and may contain any number of frames above 2. The initial left and right coordinates, as well as the velocity, are depicted in Table 3.2 for the small dataset and in Table 3.3 for the large dataset. The reduced amount of variables for the large dataset are displayed in Figure 3.2. Note that even though one of the datasets is called large, both datasets are extremely small compared to benchmark datasets. This does eliminate the need to preprocess the datasets before performing quantum annealing.

Table 3.2: Small dataset

Frame range	Index t	Left l_t	Right r_t	Velocity v_t
1-5	1	-0.1+0.08(5-F)	0.2+0.08(5-F)	0.08
	2	0.8-0.16(5-F)	0.9-0.16(5-F)	-0.16
	3	0.79	0.81	0
6-10	1	-0.1	0.2	0.05
	2	0.8	0.9	-0.1
	3	0.79	0.81	0

Table 3.3: Large dataset

Frame range	Index t	Left l_t	Right r_t	Velocity v_t
3 - ∞	1	0.75	0.85	-2/F for $f < F//2$ 2/F for $f \geq F//2$
	2	-0.1	0.2	$1/2F$
	3	0.8	0.9	$-1/F$
	4	0.79	0.81	0

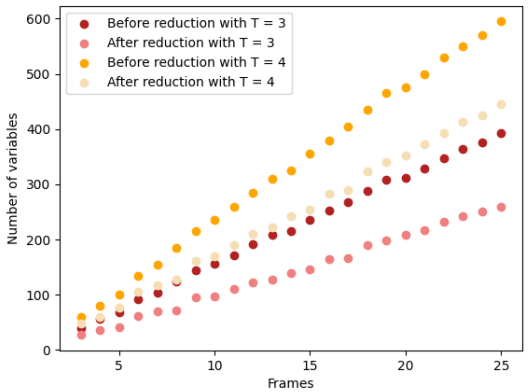

Figure 3.2: Amount of variables before and after removing unnecessary dummy variables.

0100

Quantum Annealing in Practice

Before experimenting with real quantum annealers, it is essential to discuss how quantum annealers work in practice. Quantum annealers are not found in nature, but created like computers. In 1999 the company D-Wave was founded and in 2011 they produced the *D-Wave One* [4], the first commercial quantum annealer. It contained 128 qubits and D-Wave marketed the *D-Wave One* as "quantum computer". Note that we do not view quantum annealers as computers, as you can not perform any computations on a quantum annealer. This does not mean that quantum annealers play a small role in quantum research, as both quantum computers and quantum annealers use qubits.

Over the years D-Wave has produced multiple quantum annealers, each improving on the previous generation and focussing on consumer accessibility. In 2018 D-Wave released the *Leap Quantum Cloud Service* which we also use in this report to access their quantum annealers. There are two quantum annealers that will be experimented with and discussed in this report: the *Advantage 1*, released in 2020, and the *Advantage 2*, released on the 20$^{\text{th}}$ of May 2025. These quantum annealers can be accessed through the *Leap Quantum Cloud Service*.

This report will compare and discuss multiple *Advantage* systems, such as the *Advantage_system7.1*. For these systems the first number is the system serial number and the second number is the version update.

Let us dive deeper into quantum annealing in practice. The following sections are a summary of the information from D-Wave's website [18]. In this summary we put emphasis on the differences between quantum annealing in theory, as discussed in Section 2, and quantum annealing in practice.

4.1. Types of Annealing

D-Wave provides two types of quantum annealers, or "solvers": quantum-classical hybrid solvers and quantum solvers. The hybrid solvers run on quantum processing units (QPUs) as well as on a classical computer, while the quantum solvers run purely on QPUs. In industry it currently is easier to use a hybrid solver as there are fewer parameters that the user, e.g., a company, has to feed to the solver. The hybrid part is designed to then allocate the tasks to the classical and quantum processers in an efficient manner. For a user such as a company it is not of interest which parts of their problem are solved on what type of processing units. In academia, however, it does not make sense to discuss and experiment on quantum annealers without knowing what happens within the solvers. Therefore, this report solely discusses the pure quantum solvers. A QPU from D-Wave can perform three types of annealing: standard, reversed and fast. We will discuss the standard and reversed types of annealing. See the discussion in Section 8 on fast annealing.

The Standard Annealing Path

The standard annealing path is the most similar to annealing in theory as explained in Section 2.1. The quantum system starts with a predetermined initial Hamiltonian and its corresponding ground state (Equation 4.1). The user provides the problem Hamiltonian $\hat{H}_P$. The initial Hamiltonian is fixed, therefore a quantum

system on a D-Wave annealer always starts with the following initial Hamiltonian and ground state

$$\hat{H}_I = -\frac{1}{2}\sum_i \hat{\sigma}_x^{(i)}, \quad |\phi(0)\rangle = |\phi_0\rangle_{\hat{H}_I} = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{4.1}$$

where $\sigma_x^{(i)}$ is the Pauli x operator applied to the i^{th} qubit and n is the total number of qubits. Hamiltonian $\hat{H}_I$ puts the quantum system in a maximally superpositioned state, so every qubit is equal part spin up and spin down. This initial Hamiltonian and corresponding initial ground state are similar to the ones from Farhi, discussed in Section 2.2. The problem Hamiltonian can be set by the user and must be of QUBO or Ising form. The full formula for the Hamiltonian during the annealing process then becomes

$$\hat{\mathscr{H}}(s) = A(s)\hat{H}_I + B(s)\hat{H}_P = -\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right) + \frac{B(s)}{2}\left(\sum_{i<j} \mathbf{J}_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)} + \sum_i \mathbf{h}_i\hat{\sigma}_z^{(i)}\right). \tag{4.2}$$

In Equation 2.7, and often in theory, $A(s) = 1 - s$ and $B(s) = s$. On a D-Wave solver, however

$$A(s) = \Delta_q(\Phi_{CCJJ}(s)), \quad B(s) = 2M_{AFM}|I_p(\Phi_{CCJJ}(s))|^2, \tag{4.3}$$

where the variables are physical properties as described in [19] and subject to change per annealer or annealer version. D-Wave explains that $A(s)$ represents the transverse, or tunneling, energy and $B(s)$ is the energy applied to the problem Hamiltonian. See Figure 4.1 below for the values of $A(s)$ and $B(s)$ during a standard anneal protocol of one of D-Wave's annealers.

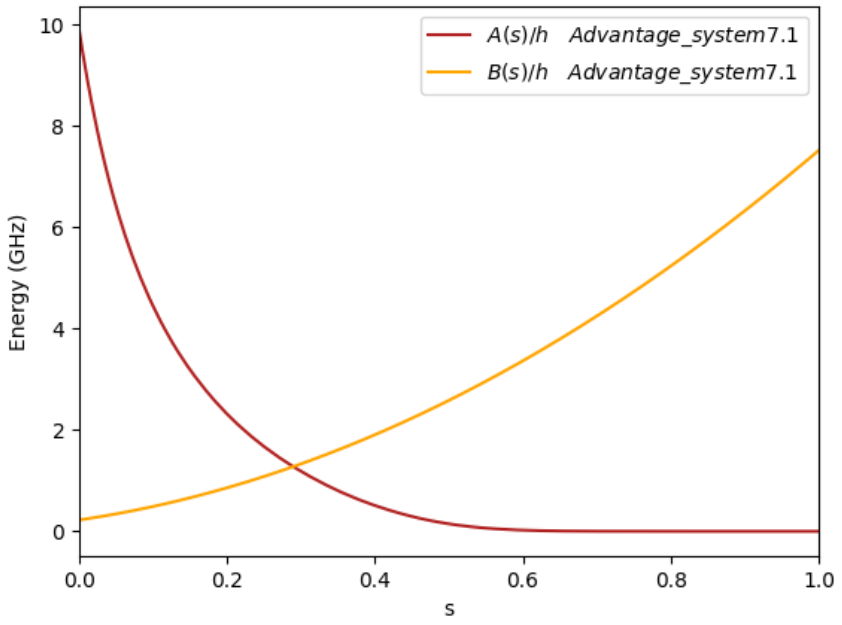

Figure 4.1: Annealing functions $A(s)$ and $B(s)$ on the D-Wave *Advantage_system7.1*. Original from [19].

Note that the system does not actually start with the initial Hamiltonian or end with the problem Hamiltonian (even after normalization). Because $A(s = 0) \gg B(s = 0)$ and $A(s = 1) \ll B(s = 1)$, we still refer to the initial Hamiltonian as $\hat{H}_I$, even though it is actually $A(0)\hat{H}_I + B(0)\hat{H}_P$. Another difference with the evolution of the Hamiltonian in theory (from Equation 2.7) is its quantum critical point.

Definition 4.1. The **quantum critical point** (QCP) is the energy at $s \in [0,1]$, such that $A(s) = B(s)$.

Because the values of $A(s)$ and $B(s)$ slightly differ per version, the QCP also slightly differs. They all are, however, not equal to $s = 0.5$ as is the case in the example from Section 2.2. Different $A(s)$ and $B(s)$ functions lead to different Hamiltonians $\hat{H}(s)$ during annealing, and therefore also to different eigenvalues of $\hat{H}(s)$. To compare the annealing process from Section 2.2 to a real-world annealing process, we have normalized $A(s)$ and $B(s)$ and calculated the eigenvalues, displayed in Figure 4.2.

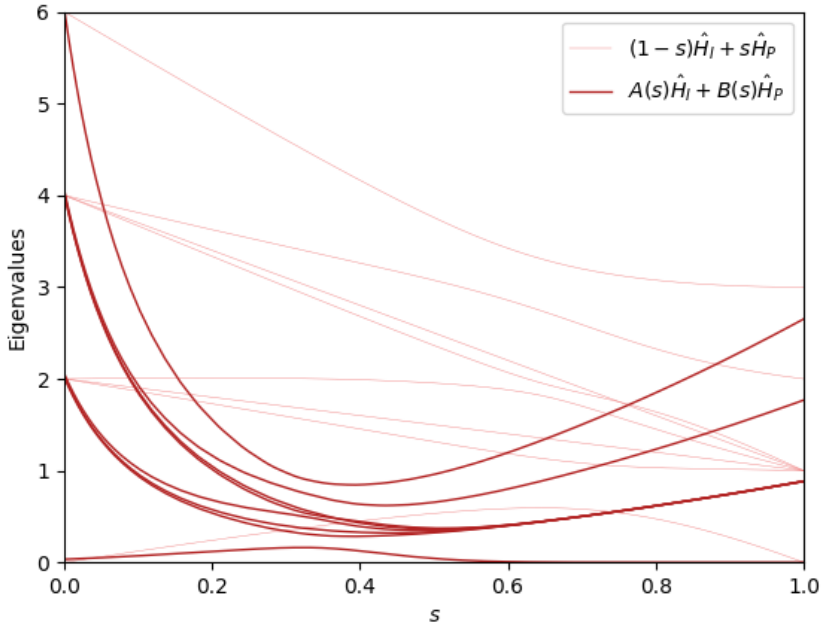

Figure 4.2: Eigenvalues on D-Wave *Advantage_system5.4* compared to the eigenvalues from Figure 2.1. Values of $A(s)$ and $B(s)$ from [19] and are scaled for comparison.

Recall that for the theoretical evolution of $\hat{H}(s)$ of the example from Section 2.2, $\delta_{\min} \approx 0.5$ and was attained at $s \approx 0.72$, whereas for functions $A(s)$ and $B(s)$, as defined in Figure 4.1, we have a minimal gap of $\delta_{\min} \approx 0.14$ at anneal fraction $s \approx 0.36$. Therefore, not only does the size of the gap differ, but where the minimum gap is attained differs significantly.

When the anneal fraction is further away from where the minimum gap is attained, the chance of exciting the quantum system to a higher energy state is smaller. Therefore, knowing where the minimum gap is attained is crucial for ensuring a successful annealing process. This is because in practice the annealing is not truly an adiabatic process, as it can never be infinitely slow on a real-world annealer. To combat this, it can increase the odds of success to slow down the annealing process only near where the minimum gap is attained. Changing the annealing speed can be done on D-Wave's quantum annealers by setting points.

On the *Advantage 1* and *Advantage 2* systems, the annealing time T can be set between 0.5 μs and 20 000 μs with a default annealing time of $T = 20\,\mu s$. It is possible for the user to only give T and $\hat{H}_P$. In that case the anneal path contains two points: $(s = 0, t = 0)$ and $(s = 1, t = T)$. The user can also set $n \geq 2$ points $(s_1 = 0, t_1 = 0), (s_2 \geq s_1, t_2 > t_1), \ldots, (s_n = 1 \geq s_{n-1}, t_n = T > t_{n-1})$. The maximum number of points that the user may specify differs per solver. The rate of change in annealing fraction s between time t_{i-1} and time t_i is $\frac{s_i - s_{i-1}}{t_i - t_{i-1}}$ and may not exceed the inverse of the minimum annealing time, which means $\frac{ds}{dt} \leq 2$, with t in μs, on the *Advantage* systems. An example of an annealing schedule with four points is displayed in Figure 4.3. If the anneal schedule contains two consecutive points where s stays the same, it is called a **pause**. If the anneal schedule suddenly ends it is called a **quench**. Because quantum annealers can not suddenly end, a quench has the largest possible rate of change of s, so $\frac{ds}{dt} = 2$. Examples of both pause and quench are displayed in Figure 4.3.

A note about the physics behind a pause and a quench: The added value of a pause is related to the behaviour of the quantum system described by the adiabatic theorem, i.e. remaining in the ground state for large values

of T. The added value of a quench is related to another edge-case of quantum system behaviour, which is when the energy of the system changes infinitely fast ($T \to 0$). When the energy of the system changes fast enough, for example by performing a quench, the quantum system is said to remain in the same state as it was before the quench [11]. Performing a quench could therefore measure the quantum system before it has a chance to reach an excited state, e.g., as a result of decoherence.

Reverse Annealing

Reverse annealing starts in the problem Hamiltonian and anneals "in reverse", i.e. starts in $s = 1$ and moves towards $s = 0$, instead of the other way around. To do this, the user has to specify the annealing fraction between 0 and 1 where the annealing stops. After the annealing process reaches this point, it moves back to $s = 1$ like it would do during standard annealing.

Reverse annealing introduces two parameters that are unavailable to other annealing types. The first parameter is the initial state of the system. Where the initial state is fixed to be the maximally superpositioned state from Equation 4.1, it can be set to any classical state for reverse annealing. This way you can input a feasible solution to the problem and see if reverse annealing produces a better (feasible) solution.

The user specifies the problem Hamiltonian as well as an initial classical state of the system. The annealing process then starts on $t = 0$, at $s = 1$ with $\hat{H}_P$ and in the initial state set by the user. Furthermore, the user specifies a point s_r between 0 and 1, such that the annealing fraction decreases from $s = 1$ to $s = s_r$ and then increases again and ends with $s = 1$. See Figure 4.4 for an example of an annealing schedule for reverse annealing.

Note that the initial state must be a classical state, as it is not possible to perform quantum operations, or gates, on the qubits. This should not be a problem, as optimization problems used for quantum annealers contain only classical states as feasible solutions.

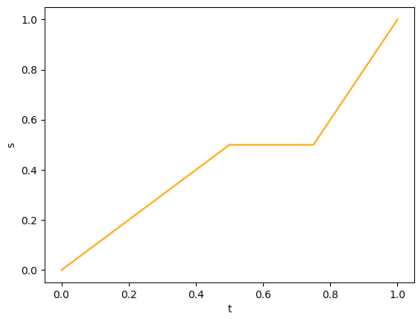

Figure 4.3: Standard annealing schedule with four points: [(0, 0), (0.5, 0.5), (0.5, 0.75), (1, 1)].

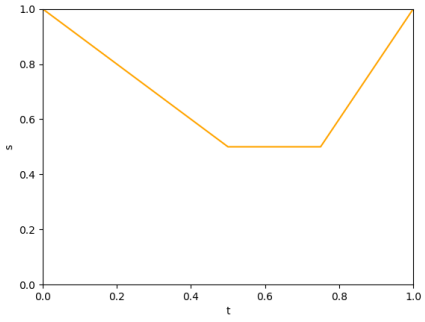

Figure 4.4: Reverse annealing schedule with four points: [(1, 0), (0.5, 0.5), (0.5, 0.75), (1, 1)]. Here $s_r = 0.5$.

The other parameter that can be set with reverse annealing determines what happens with the output, i.e. the classical state measured after the annealing process. It is possible to start every run with the same initial state, but it is also possible to use the outcome from the previous run as the initial state for the next run. This way you could use multiple reverse annealing runs to (hopefully) attain better solutions.

4.2. Quantum Processing Unit

The quantum annealing is done on a quantum processing unit, or QPU. D-Wave has developed different QPUs and also refers to them as solvers. Before testing on quantum annealers/QPUs/solvers, we will discuss some of the key features of QPUs and how they work. The *Advantage 1* has multiple versions. We only use the *Advantage_system7.1*, which is the newest version of the *Advantage 1* available to us. At the start of this project, only the prototype of the *Advantage 2* was publicly available. On the twentieth of May 2025, however, D-Wave released the *Advantage2- _system1.1* to the public. Because this was during testing and writing this report, the *Advantage2_prototype2.6* was also used at the start.

Differences across solvers include the number of qubits, temperature, layout of qubits, denoising procedures and more. Not all of this information is publicly available (see Section 6.3). What is available is how the qubits are laid within the QPU. Because perfectly performing qubits do not exist (yet), it is of interest how the qubits are laid in the QPU and how they are connected to each other, which is called the topology.

Topology

Both the *Advantage 1* and the *Advantage 2* use superconducting flux qubits [5], but utilize completely different topologies [20]. The topology of the *Advantage 1* is called Pegasus. A small part of the Pegasus hardware graph is displayed in Figure 4.5. The *Advantage 2* builds upon the Zephyr topology, as displayed in Figure 4.6.

In the Pegasus topology, every qubit has 12 neighbours, and it is possible to construct a K_4 graph, which is the graph where four qubits are all directly connected to each other. In the Zephyr topology, every qubit is connected to 18 other qubits, and it is possible to construct a complete K_4 graph, which is when 4 vertices are all direct neighbours of each other, as well as a complete bipartite graph $K_{8,8}$ [21, 20]. Before D-Wave developed the Pegasus and Zephyr topologies, the Chimera topology was used. Here every qubit was connected to 6 other qubits and K_4 subgraphs could not be embedded without using chains (which will be explained later).

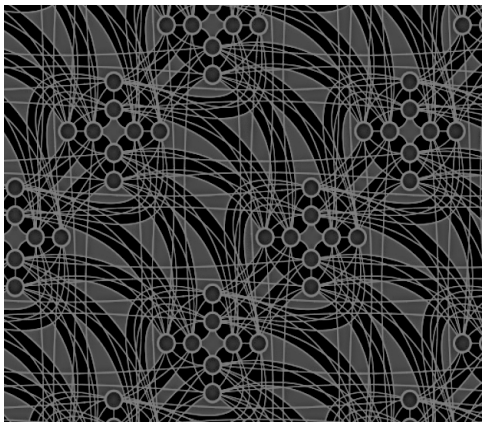

Figure 4.5: Screenshot of Pegasus topology. The circles are qubits. Light gray lines between qubits are couplings, making two qubits neighbours.

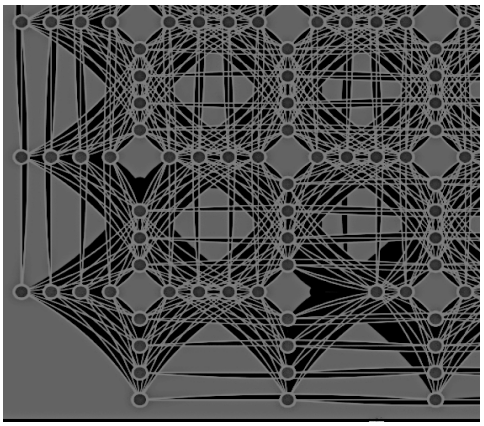

Figure 4.6: Screenshot of Zephyr topology at the edge of the QPU. The circles are qubits. Light gray lines between qubits are couplings, making two qubits neighbours.

Note that because of the challenges related to qubits, it may be the case that not all qubits and all couplings are working in a concrete QPU instance. The qubits and couplings that are working form the **working graph**. The working graph may change every time the system is initializes the quantum state for a new run or after system updates. D-Wave does not share all information on the working graph, so we will assume the working graph is close enough to the full topology to not interfere with the quantum annealing runs we will perform.

Embedding

We established that most graphs can not directly be translated to the qubits within a QPU, as qubits within the QPU are not fully connected. Therefore, we have to take into account how the qubits from the Ising model are related to the physical qubits in a QPU. For readability, qubits from the original, or input, Ising model will be referred to as variables, and the term qubit will exclusively be used when talking about real physical qubits. To run a problem on a quantum annealer, the problem has to be embedded onto the QPU, which is the assignment of variables from the input problem to qubits within their topology.

There are a couple rules to correctly embed an Ising model onto qubits of a QPU. Every variable has to have at least one qubit assigned to it, because the measurement outcome of the qubits after annealing determine the solution to the Ising or QUBO problem. Additionally, the qubits have to be chosen in such a way that if two variables are connected, meaning $\mathbf{J}_{ij} \neq 0$, then the qubits assigned to variables i and j also have to be connected with an edge weight, also called coupling strength, equal to (normalized) $\mathbf{J}_{ij}$.

Recall that the currently employed Zephyr topology has qubits with 18 neighbours. That means that if one

were to embed a variable v with 19 or more neighbours, one could not find an embedding where one qubit represents variable v. To combat this, the embedding uses **chains**. A chain is when multiple qubits are used to represent the same variable, e.g., a variable v with 19 neighbours. If the chain consists of two qubits, they in turn have to be connected by an edge. If more than two qubits represent the same variable, they must have a path between them to form a chain. A (part of a) chain is simply an edge between two qubits and is visualized on the D-Wave UI by a thick light gray edge (see Figures 4.7 and 4.8).

In conclusion, to have an acceptable embedding of a given Ising problem, every variable has to be translated to a qubit, or chain of qubits, such that every pair (u, v) of connected variables is represented by one (or more, see Section 6.2) physical coupling(s) between (at least) one of the qubits representing variable u and (at least) one of the qubits representing variable v.

Finding a correct embedding, especially when minimizing the amount of qubits, is its own optimization problem. See Figures 4.7 and 4.8 for an embedding of a 27-variable MOT problem on the Pegasus and Zephyr topologies using 49 and 47 qubits, respectively. Note that because of the chains, one may need significantly more qubits of the QPU than the amount of variables. If an embedding requires more qubits than the QPU contains, the problem simply can not be embedded and therefore not be solved using the QPU at hand.

Figure 4.7: MOT instance of 27 variables embedded onto 49 qubits in Pegasus topology. Chains are in light gray. Measurement of one sample of the system is displayed on the qubits with circle filling blue corresponding to +1 and white to -1.

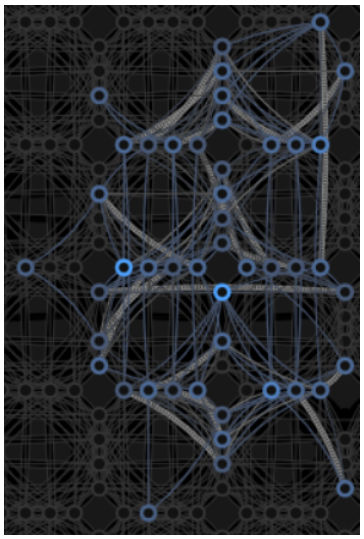

Figure 4.8: MOT instance of 27 variables embedded onto 47 qubits in Zephyr topology. Chains are in light gray. Visualisation by D-Wave UI without showing sample measurement

Because the chains are real couplings between qubits, they must contain a weight like any other edge in the embedding. The weight of the chain edges is called the **chain strength**. The chain strength can be set by the user, or be set to the default value by D-Wave. After completion of the annealing process, the qubits in the chain are all measured to be 1 or -1. The value of the variable is then equal to the majority of the measured values. If not all qubits in the chain have the same measurement outcome, the chain is called broken (which can also be shown in the D-Wave UI). To ensure all qubits in the chain are measured to have the same outcome, the couplings within the chain have to be strong enough, i.e. the chain strength must be chosen large enough, but with negative sign as we want to minimize [22]. Similar to the Lagrangian multiplier, if the chain strength is too large the rest of the Ising model becomes insignificantly small and loses its effect on the annealing process.

How the chain strength is set is explained in Section 5 and how it affects the quantum annealing results is part of the experiments in Section 6.4.

Output

After the annealing is complete, the qubits are measured to be -1 or +1. The measurement of the system is called a sample. D-Wave automatically translates the measurement to the corresponding variable values,

which are -1 or +1 if the user gave an Ising model as input and 0 or 1 if the user gave the input in QUBO form. D-Wave also returns the corresponding energy or objective function value of the input problem. This does not incorporate the chain couplings. The aim is to achieve the sample that produces the ground state energy. If the system excites during annealing the sample will correspond to a higher energy. Therefore, keep in mind that the lower the energy of the results, the better. To properly test everything, we perform a thousand runs and measure a thousand samples.

0101

Initial setup and results

To gain a deeper understanding of quantum annealing, we test our dataset from Section 3.3 on the quantum annealers of D-Wave. In Section 4 we showed that quantum annealing is significantly more complicated in practice than in theory (Section 2). D-Wave offers a lot of ways to influence the annealing process. Because quantum annealing is still a new field, it remains unknown what the best parameters are. First we fix the instances of the dataset and the solvers that we want to test. Then we provide the corresponding Ising models to a heuristic that embeds the Ising models/original problems from the dataset instances in the qubits from the QPUs. After that, we use simulated annealing to find a suitable Lagrangian multiplier for each dataset instance and discuss the initial performance of the quantum annealers.

Dataset Instances and Lagrangian Multiplier

We test five instances of our dataset on three solvers: simulated annealing, the *Advantage_system7.1*, and the *Advantage2_prototype2.6*. The five instances are chosen such that the instance with the largest n is too large to be embedded onto the *Advantage2_prototype2.6*. We only look at the reduced amount of variables, n, so the original amount of variables is not used from now on. All five instances are defined in Table 5.1.

Table 5.1: Five dataset instances and their reduced number of variables n.

Instance	Name	Frames	Tracks	Variables	n
1	Small	2	2	15	8
2	Small	3	3	27	27
3	Large	4	4	80	60
4	Large	9	4	215	162
5	Large	10	4	235	169

To finish defining the Ising models for the experiments, we need to set the penalty parameter, which is the Lagrangian multiplier, or λ (see Section 3.2). This is done by performing simulated annealing for different values of λ and choosing a value that is just large enough such that the simulated annealer returns mostly feasible solutions and the returned solutions with the lowest energy are feasible solutions. This process is explained in detail for instance 1.

For instance 1, with $n = 8$, $\lambda = 0.2$ was chosen. We also performed simulated annealing for $\lambda = 0.1$ and $\lambda = 0$, i.e. no penalty. For each λ a thousand rounds of simulated annealing were performed. See Figures 5.1, 5.2 and 5.3 for the simulated annealing results of λ equal to 0, 0.1, and 0.2, respectively. Here, $\lambda = 0.2$ is a good choice because the lowest energy value solution is also feasible, which is not the case for values of λ below 0.2 that were tested. Note that, because we measure a thousand times on quantum annealers, we also do a thousand runs for simulated annealing. The high amount of measurements leads to a log-scaled y-axis in all resulting histograms.

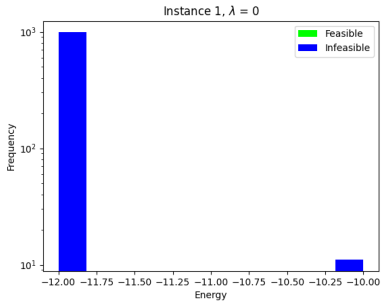

Figure 5.1: Simulated annealing without penalty parameter.

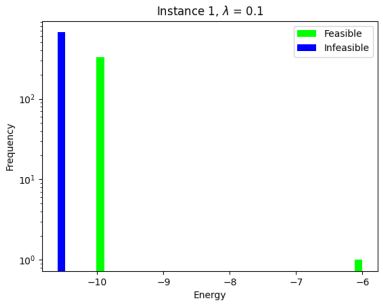

Figure 5.2: Simulated annealing with λ too small.

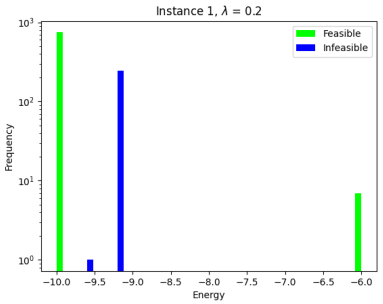

Figure 5.3: Simulated annealing with large enough λ.

To find the Lagrangian multiplier for instance 2 we increased λ until we got satisfying results (see Figure 5.4). That is, a majority of lower energy value solutions are feasible. This was then done for instance 3 (Figure 5.5), 4 (Figure 5.6), and 5 (Figure 5.7) as well.

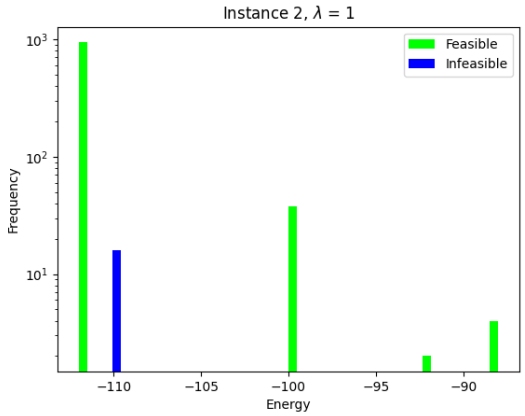

Figure 5.4: Simulated annealing on instance 2 with $\lambda = 1$.

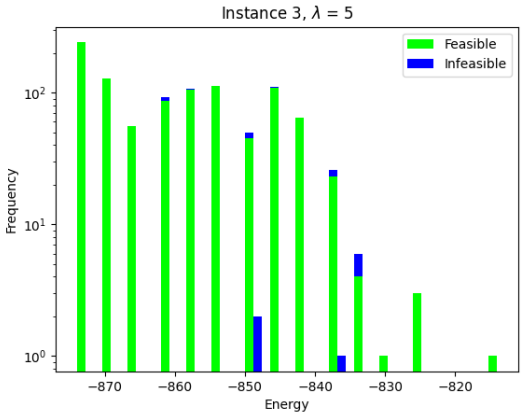

Figure 5.5: Simulated annealing on instance 3 with $\lambda = 5$.

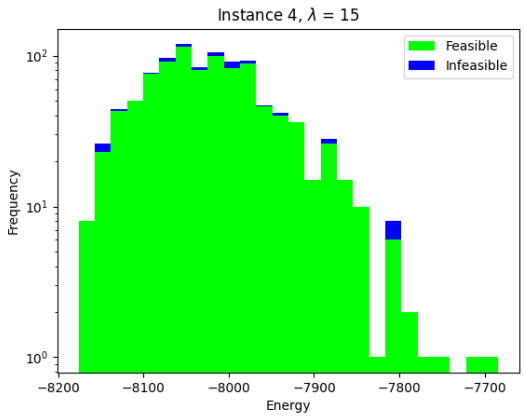

Figure 5.6: Simulated annealing on instance 4 with $\lambda = 15$.

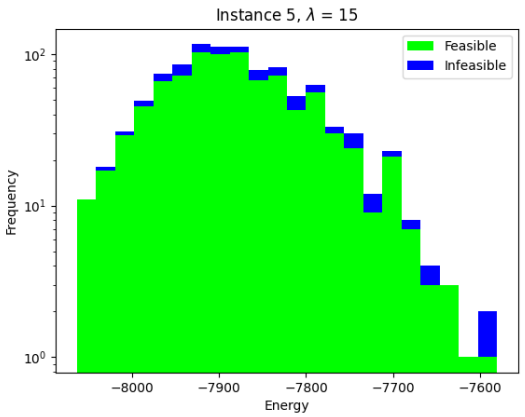

Figure 5.7: Simulated annealing on instance 5 with $\lambda = 15$.

Note that the more variables there are, the more different values of energy are measured/found. To keep the results more readable we set the amount of bins in the histograms to 50. With these settings some of the infeasible and feasible solutions are binned together while having different energy values. This does not affect the results unless this happens for the lowest energy bin, in which case we also plot the results using more bins.

Table 5.2: Parameters of instances 1 through 5, and their corresponding λ and chain strength. Range of number of qubits of embedding the instances on Pegasus and on Zephyr topologies.

Instance	n	λ	# qubits Pegasus	# qubits Zephyr	Rounded chain strength
1	8	0.2	8	8-9	2.925
2	27	1.0	49-59	47-53	8.218
3	60	5.0	187-232	164-184	33.109
4	162	15	1255-1524	899-1043, N.A.	105.140
5	169	15	1325-1506	N.A.	103.611

Embeddings, chain strength and number of runs

To perform quantum annealing on these Ising models, the variables are mapped onto qubits of the QPUs of the solvers. As discussed in Section 4.2, it is hard to find a good embedding. A separate embedding is needed for each instance and for each solver. To find the embedding, the heuristic [16] from minorminer [23] was used 10 times, with random seed 0, 1, ..., 9. The embedding with the least amount of qubits was saved and used for every test done on its corresponding instance and solver. For the range of the amount of qubits used in the embedding see Table 5.2. Instance 5 could not be embedded onto the *Advantage2_prototype2.6* solver, as it does not have enough qubits, and therefore can not be tested.

Note that the embeddings found by minorminer show a large variation in their number of qubits. Additionally, it could take up to half an hour to find an embedding for instances 4 or 5. There is ample opportunity for improvement, as the heuristic is built for sparse $\mathbf{Q}$ and does not take into account the newer topologies of the solver, like Pegasus and Zephyr. Whether the embedding affects the results will be researched in Section 6.3.

To perform quantum annealing we used the default chain strength, which is found by a built-in formula that "attempts to compensate for chain-breaking torque" [24]. The built-in, or default, chain strength does not depend on the embedding or the solver. The Lagrangian multiplier was kept the same as found for simulated annealing and every (possible) combination of instance and solver was performed for a thousand runs with T set to the maximal annealing time of 2000 μs.

Results

Now that all parameters are defined, we can perform quantum annealing on real D-Wave solvers. To do a thousand runs without exceeding the maximally allotted solver calling time of one second, the solvers were called three times: first with 350 runs, again with 350 runs and lastly with 300 runs. This should not affect the results. The total runtime for a solver on a given instance was around 2.2 seconds.

For instance 1, see Figure 5.8 for the performance of the *Advantage_system7.1*, and Figure 5.9 for the performance of the *Advantage2_prototype2.6*. On both solvers the problem of $n = 8$ could be embedded onto 8 qubits, so the (default) chain strength does not influence the results. We compare the results to the simulated annealing results from Figure 5.3. Both solvers perform well, that is return the same lowest feasible state as the simulated annealer. The *Advantage2_prototype2.6* performs a bit better and returns the lowest feasible state more often.

For instance 2 the chain strength does play a role as the solvers need almost twice the amount of qubits to embed the 27-variable problem. The embeddings of this exact instance are displayed in Figures 4.7 and 4.8. The results of a thousand measurements/runs are displayed in Figures 5.10 and 5.11 and again have comparable results to each other and to the simulated annealing results from Figure 5.4. The same second-to-lowest feasible state as was found by simulated annealing was attained on both solvers, but more often on the *Advantage_prototype2.6*.

For instance 3, the *Advantage_system7.1* slightly outperforms the *Advantage2_prototype2.6*, but both solvers measure significantly more infeasible and higher energy states than the simulated annealer from Figure 5.5.

At instance 4 the problem size starts to scale up. With 162 variables and embeddings of 899 (Zephyr) or 1255 (Pegasus) qubits, both annealers perform much worse than on previous instances. The *Advantage_system7.1* performs better than the *Advantage2_prototype2.6* with one feasible solution and with more results with lower energy and/or that are feasible. However, both solvers perform terribly compared to the simulated annealing results from Figure 5.6.

For instance 5 we could only use the *Advantage_system7.1* annealer. The performance is slightly better than for instance 4, but still nowhere near the performance of the simulator from Figure 5.7. Even though instance 5 has 7 more variables than instance 4, the default chain strength is slightly lower. This may explain the better results, but this is not enough information to make such conclusions. That is why we look more into the chain strength in Section 6.4.

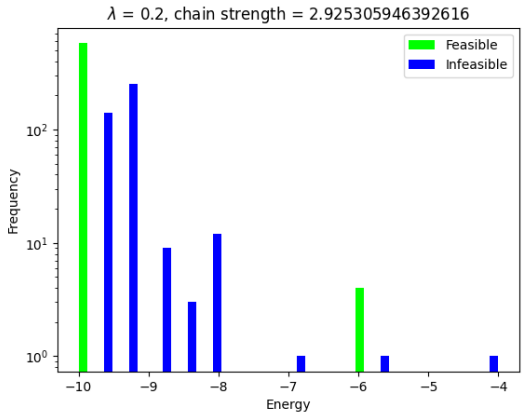

Figure 5.8: Results for instance 1 with 1000 runs of quantum annealing on the *Advantage_system7.1* solver.

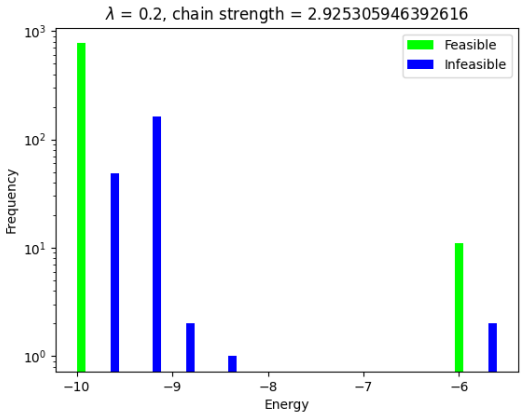

Figure 5.9: Results for instance 1 with 1000 runs of quantum annealing on the *Advantage2_prototype2.6* solver.

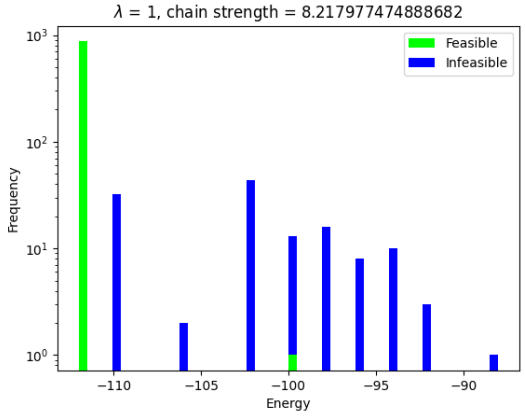

Figure 5.10: Results for instance 2 with 1000 runs of quantum annealing on the *Advantage_system7.1* solver.

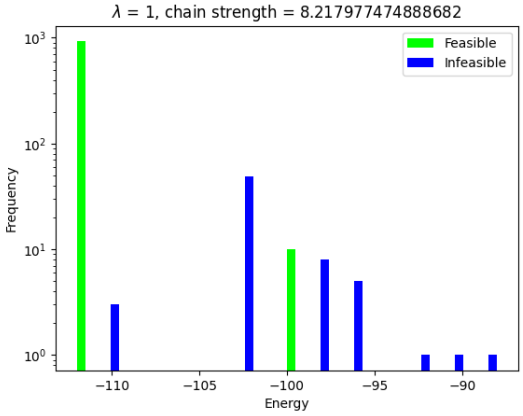

Figure 5.11: Results for instance 2 with 1000 runs of quantum annealing on the *Advantage2_prototype2.6* solver.

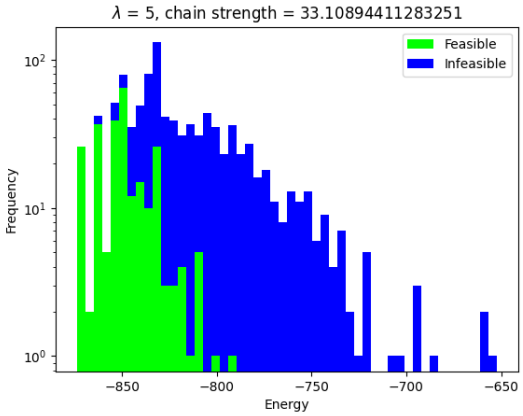

Figure 5.12: Results for instance 3 with 1000 runs of quantum annealing on the *Advantage_system7.1* solver.

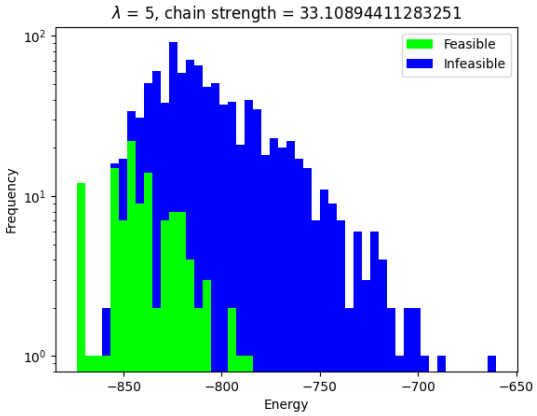

Figure 5.13: Results for instance 3 with 1000 runs of quantum annealing on the *Advantage2_prototype2.6* solver.

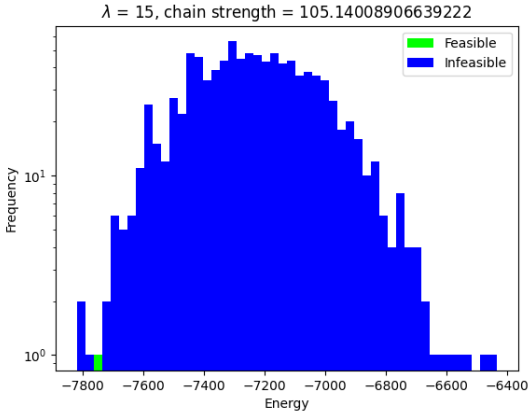

Figure 5.14: Results for instance 4 with 1000 runs of quantum annealing on the *Advantage_system7.1* solver.

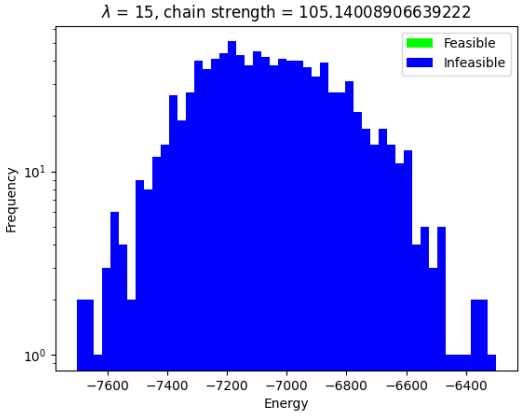

Figure 5.15: Results for instance 4 with 1000 runs of quantum annealing on the *Advantage2_prototype2.6* solver.

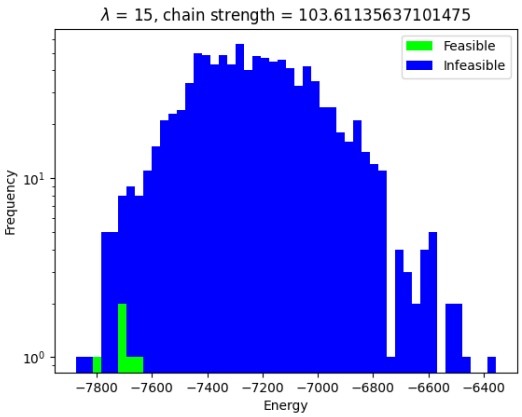

Figure 5.16: Results for instance 5 with 1000 runs of quantum annealing on the *Advantage_system7.1* solver.

0110

Parameter and Anneal Schedule Optimization

It is safe to assume that the default parameter values and values found using simulated annealing are not the optimal parameters. The results from the previous section serve as a performance baseline, which we will try to improve upon in this section. For every quantum annealing experiment we will perform a thousand runs and consider the results to be an improvement if more and/or better feasible solutions are found. Note that we wish to minimize the objective function, or Ising model energy, therefore a "better" feasible solution has a lower objective function (and Ising model energy) value. Note that it is also possible to get lower energy value with infeasible solutions, but we do not take these into account. If too many of the lower energy solutions measured are infeasible, however, it could mean that the penalty parameter λ is too low, i.e. infeasible solutions are not penalized enough.

We will focus on instance 4, which has 9 frames, 4 tracks, 162 variables, and at least 899 qubits (see Table 5.1), as there is ample room for improvement with only one feasible solution found during the initial testing done in Section 5 (see Figures 5.14 and 5.15).

There are multiple aspects of annealing that could improve the results. For example the annealing time T and the chain strength cs, which will be researched in Section 6.2 and Section 6.4, respectively. To optimize the parameters we choose one of two methods that are introduced and explained in Section 6.1. During these experiments other results are also found, which relate to parameters beyond our control. These results give a crucial insight in the stability of the performance of D-Wave's QPUs and are discussed in Section 6.3. The findings from Section 6.3 are so crucial that they will be taken into consideration for remaining experiments. Additionally, we try to achieve better performance by adjusting the anneal schedule and by testing reverse annealing in Sections 6.5 and 6.6, respectively.

6.1. Method

It is most likely that the values for penalty parameter λ, found with the help of simulated annealing in Section 3.2, maximal annealing time T, and default value of the chain strength parameter cs are not optimal for real-world quantum annealers. Research has been done on suitable values of these parameters, but for simulated annealing [25]. Here, because we have access to QPUs we want to find suitable parameters for specific QPUs and for quantum annealing in general. To do this we tried the following two methods.

Method 1: Performing simulated annealing on the embedding of the Ising model.

The quantum annealers showed worse performance than the simulator. This could be a cause of the chain structure and its corresponding chain strength, as there are no chains in the simulation. It is possible, however, to incorporate the chains on the simulator, by incorporating the same embedding as used for quantum annealing. To achieve this the problem is transformed from an n-variable QUBO to an q-variable QUBO

where q is the amount of qubits of an embedding. The chains are then part of the QUBO like the rest of the objective function.

To transform the n-variable problem into a q-variable problem, we first look in more detail at how D-Wave creates these chains. The embedding made using the minorminer heuristic is available to see how the chains are created. In Figures 6.1 and 6.2 we see variable 9 and 10 of instance 2 embedded onto the Zephyr topology. Variable 10 is embedded onto the chain highlighted in Figure 6.1 and variable 9 is connected to 10 via the two highlighted edges from Figure 6.2. The original edge weight between variables 9 and 10 is equal to 2. In the embedding variables 9 and 10 are connected via two edges, both with edge weight 1.

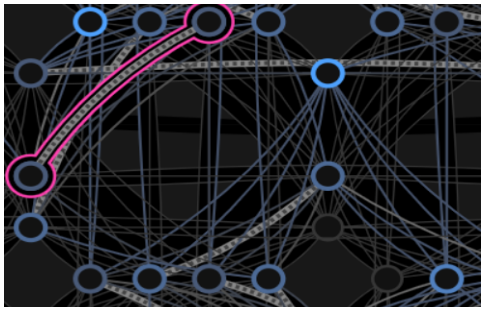

Figure 6.1: Screenshot of the embedding. Variable 10 consists of two qubits and is highlighted in pink.

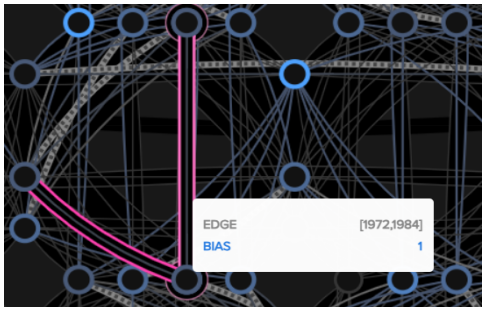

Figure 6.2: Screenshot of the embedding. Variables 9 and 10 are connected by the two couplings that are highlighted in pink.

We have observed the same for other qubit chains. Therefore, if two variables are (both) embedded onto chains they may be connected with more than one edge/coupling. In that case both couplings are used with lower coupling strength such that their sum equals the original edge weight. Note that this is based on our observations of D-Wave's user interface instead of being based on documentation. This was mimicked when transforming the original problem into the embedded problem in a way that simulated annealing could be performed. The new problem has the exact same structure and edge weights as in the embedding, including the chain strength between qubits that are connected through a chain. The simulated annealer does not distinguish chain strength from coupling strength, meaning chains are now part of the new problem/Ising model like any other edge weight. Whether, or how the QPU makes the distinction between coupling strength set by the Ising model or by the chain strength is unknown (to us at this time). After transforming the problem, the simulated annealing heuristic is performed on this transformed problem. The results of instances 3 and 4 are depicted in Figures 6.3 and 6.4, respectively.

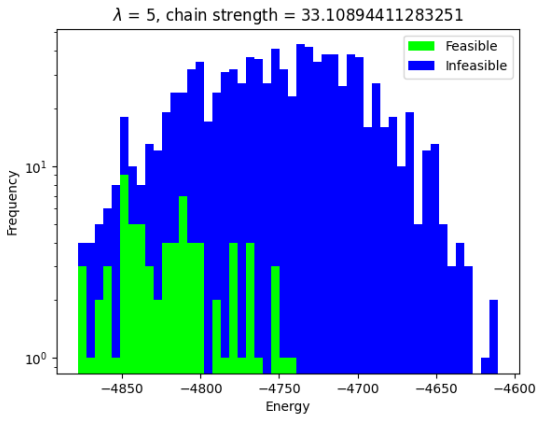

Figure 6.3: Results for instance 3 with 1000 runs of simulated annealing on Zephyr qubit topology.

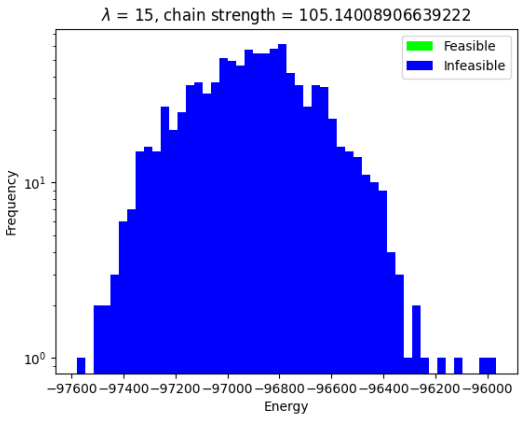

Figure 6.4: Results for instance 4 with 1000 runs of simulated annealing on Zephyr qubit topology.

For smaller instances, with relatively few chains, the simulated annealer performs roughly the same on the new and the original problem. For larger instances, the simulated annealer fails to produce feasible results.

This is a similar performance to the preliminary quantum annealing experiments from Section 5, but significantly lower than the performance of the *Advantage 2*, which will be discussed later. This bad performance may be a result of the built-in normalisation of the edge weights/couplings that most likely is different between quantum annealing and simulated annealing. The simulated annealer could not distinguish between chain and non-chain couplings, which could have led to the high amount of broken chains. Instead of adjusting the normalisation which could make it harder to compare simulated annealing to quantum annealing, we decided to use another approach to find a suitable chain strength.

Note that this method of combining the qubit embedding with the simulated annealing solver is not found to have been employed in literature. We still think this method could benefit the understanding and the performance of simulated annealing as well as quantum annealing, but this would require a better understanding of the normalization of quantum and simulated annealing as well as more parameter testing, which is beyond the scope of this thesis.

Method 2: Grid search.

The second method to achieve better quantum annealing performance is to perform grid search for the parameters and document their impact on the annealing results. This method does not rely on simulated annealing and its subtle differences with quantum annealing. The downside of this method is that it requires many quantum annealing runs to get conclusive results, and therefore costs a lot of annealing time.

6.2. Annealing Time

The annealing time T can be set between 0.5 and 2000 microseconds on the *Advantage* solvers with a default annealing time of 20 microseconds. The initial tests from the previous section were all performed with the maximal annealing time of 2000 μs, since the adiabatic theorem states that T must be "large enough". While this in theory gives lower odds of exciting into a higher energy state and therefore better performance, in practice the qubits also experience noise, which leads to decoherence and can significantly excite or otherwise change the quantum system during annealing. The higher the annealing time, the higher the odds of noise affecting the system, so T can not be "too high". The goal therefore is to find an annealing time T that is "high enough" but not "too high".

As it is unclear what a good value for T would be, we first test for T equal to 0.5 μs, 20 μs, 50 μs, as well as for all T that are a multiple of 100 μs up to the maximal of 2000 μs. To determine the performance we looked at the amount of feasible solutions and their energies compared to other energies measured. Note that the initial experiments of Section 5 were performed on the *Advantage_system7.1* and the *Advantage2_prototype2.6*, where the upcoming results will be from the brand new *Advantage2_system1.1* (see Appendix A). The embedding of instance 4, found for the *Advantage2_prototype2.6* (899 qubits) can not be transferred to the *Advantage2_system1.1*, even though the newer solver has the same topology and more qubits. Therefore, we had to again find embeddings using 10 different random seeds which resulted in an embedding of 977 qubits.

All other parameters were kept the same and are the following: Instance 4, λ set to 15, default chain strength cs of around 105 (see Table 5.1). Then for every annealing time T we performed a thousand runs and kept track of the feasible solution with the lowest energy measured. To later compare with different λ values, we translated the energy of the sample to the energy of the Ising model with no Lagrangian multiplier. For example, for $T = 2000\,\mu s$ the minimal feasible energy (without λ) measured is -168. With $\lambda = 15$ the measured energy of this same sample is -8088, which is the lowest energy measured from the thousand runs performed with $T = 2000\,\mu s$. The full histogram is shown in Figure 6.5. Such a histogram was made for every annealing time T, which would be too much to display. Here, we are only interested in the lowest energy without λ attainable from the feasible samples measured. The other histograms are available in the Github repository (see [26]). The best feasible solutions measured for different annealing times T are displayed in Figure 6.6.

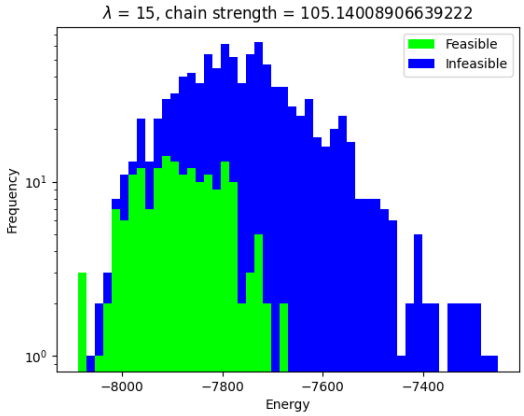

Figure 6.5: Quantum annealing results of a thousand runs performed on instance 4, with parameter values $\lambda = 15$, default $cs \approx 105$, and $T = 2000\,\mu s$.

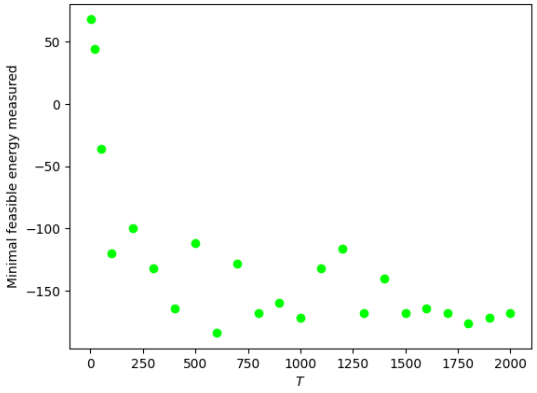

Figure 6.6: Feasible solution with the lowest objective function value taken from a thousand annealing runs per annealing time T value. Annealing done on instance 4 with $\lambda = 15$ and default chain strength $cs \approx 105$. The best performing sample is at $T = 600\,\mu s$ with an objective function value of -184.

Note that the histogram from Figure 6.5 is generated from the same experiment as in Figure 5.15, meaning both have the same Ising model and the same values for T, λ, chain strength, and topology. The only difference is that the results from Figure 5.15 are from the *Advantage2_prototype2.6* annealer and the new results from Figure 6.5 are from the *Advantage2_system1.1* annealer. The results of the new solver are already drastically better, without having done any parameter optimization, as instead of 0 there were 178 feasible samples measured. This is to be expected, as all the feasible samples have an energy of at least -7600 (x-axis value in both figures), where the *Advantage2_prototype2.6* solver attained almost no samples with such a low energy. In other words, all samples found with the *Advantage2_prototype2.6* solver were excited to states of higher energy such that we could no longer find feasible samples.

As for the results for different annealing time values, Figure 6.6 suggests that the performance improves (so lower y-values) for longer annealing time. The results from Figure 6.6 look promising enough to further test on annealing time values between 1700 and 1800 μs as well as some extra values such as $T = 450\,\mu s$. See Figure 6.7 for the additional results.

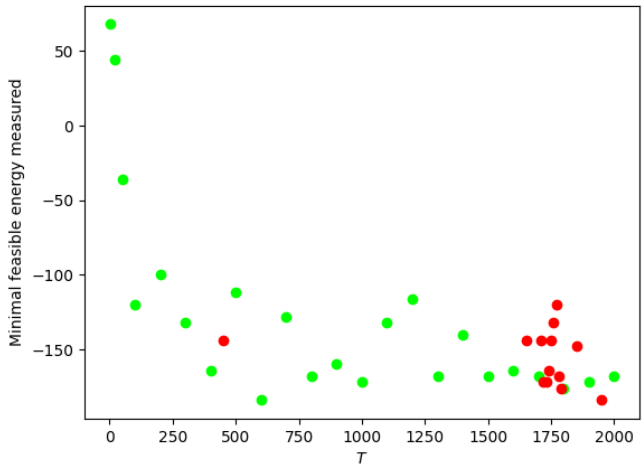

Figure 6.7: Feasible solution with the lowest objective function value taken from a thousand annealing runs per annealing time T value, with additional values in red. Annealing done on instance 4 with $\lambda = 15$ and default chain strength $cs \approx 105$. The best performing samples are at $T = 600\,\mu s$ and $T = 1950\,\mu s$, both with an objective function value of -184.

The performance of quantum annealing still seems to improve when the annealing time increases, but with lower odds than we initially expected. This means that the good results from Figure 6.6 where more of a coincidence, where it first seemed to be a direct result of the annealing time T. Note that the added annealing time, e.g., total annealing time used over the course of one month, is a finite resource. Therefore, it is beneficial to not set the annealing time T parameter longer than necessary, that is not increase the annealing time T when it most likely does not improve performance. To balance the output caused by the annealing time T parameter without using unnecessary amounts of the total annealing time resource, we recommend to set the annealing time T to 600 μs for this MOT Ising model and the *Advantage 2* QPU. If available annealing time is not a scarce resource, we recommend using higher T. Note that this annealing time may affect the results in a totally different way for newer solvers. It is expected that newer solvers have better denoising methods, in which case a higher annealing time might produce better results, or at least with a higher probability. This probability could be very low, however, as these are already the best results from a thousand runs.

6.3. Hidden Annealing Parameters

The measurements performed in this research are experimental in the sense that we employed methods such as grid search without being able to anticipate the outcome, as it is not known how MOT problems compare to other QUBOs discussed in literature. During this time D-Wave released the *Advantage2_system1.1*. This affected the performance of the *Advantage_system7.1* as it was being decommissioned in the course of the project. See Appendix A for the timeline of the testing performed over the course of the project as well as the different QPUs accessible.

During the testing in Section 6.2 we found multiple results unrelated to the parameters described previously. Here are some of the results that were discussed previously, but with more context.

Day to day differences

During testing it sometimes was unclear why the annealing performance seemed to drastically change from one experiment to the next. For example, we first started testing on the *Advantage_system7.1* and the *Advantage2_prototype2.6*, and already employed part of the grid search to achieve better performing parameters. For example, the results of Figures 6.8 and 6.9 are of the *Advantage_system7.1* using the same parameter values, and were acquired on consecutive days.

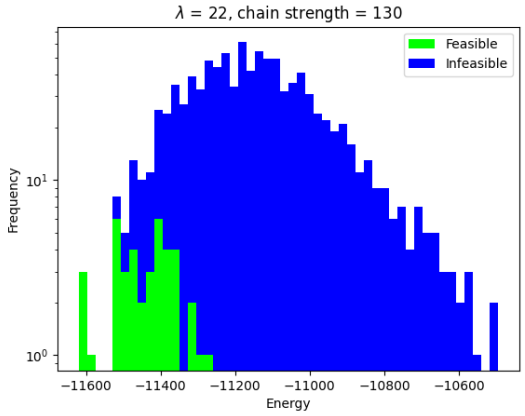

Figure 6.8: Quantum annealing measurements on the first day taking from a thousand runs performed with the *Advantage_system7.1* on instance 4 with parameter values $\lambda = 22$, $cs = 130$ and $T = 600\,\mu s$.

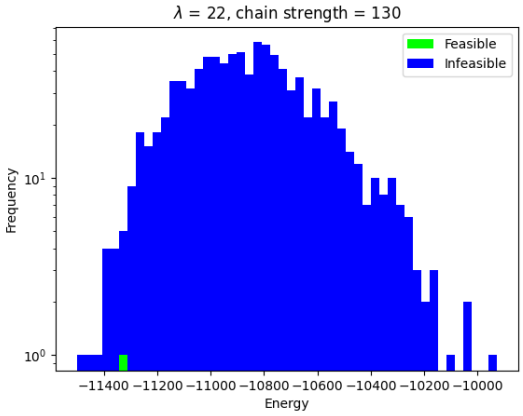

Figure 6.9: Quantum annealing measurements on the second day taking from a thousand runs performed with the *Advantage_system7.1* on instance 4 with parameter values $\lambda = 22$, $cs = 130$ and $T = 600\,\mu s$.

On the second day the performance was significantly lower. This was shortly before the *Advantage_7.1* solver was decommissioned and directly after the release of the *Advantage 2* system. We observed a deterioration in performance of the *Advantage_system7.1* QPU in these experiments and other experiments done on and after the second day. Therefore, all following experiments presented in this report will use the *Advantage 2*.

Additionally, the performance of the new *Advantage2_system1.1* seemed to also change per day. To test if this was true we performed the same experiments from the previous section again on the following day. Here, the results are visualized in the same way as on the first day (see Figure 6.7), so keeping track of the best feasible solution measured for different annealing times. The results from the previous section as well as the results from the identical experiments that were performed on the following day are displayed in Figure 6.10.

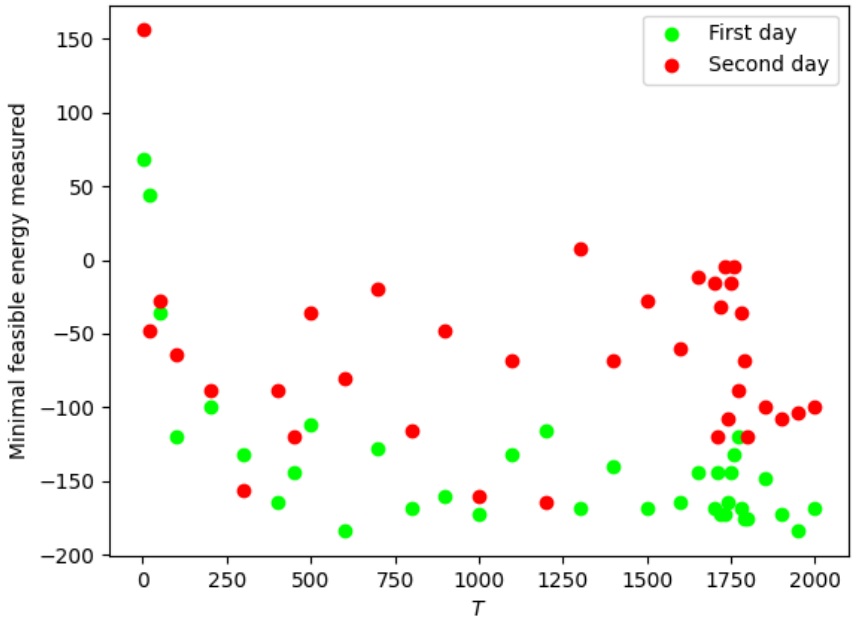

Figure 6.10: Feasible solution with the lowest objective function value taken from a thousand annealing runs per annealing time T value, with results from the second day in red. Annealing done on instance 4 with $\lambda = 15$ and default chain strength $cs \approx 105$. The best solution from the second day produces an objective function value of -164 and is attained at $T = 12000\,\mu s$.

It is clear that the results from the second day are worse than those of the first day, as the best feasible results from the second day have higher objective function value. Additionally, for some T not even one of the thousand samples was feasible. It is curious that the second day's performance was consistently worse than the day before, as this means the quantum annealing results are also affected by uncontrollable external influence factors. Note that such external influence factors are unknown. For example, it may be the case that the *Advantage 2* system was still being calibrated, as it was just launched, and that this affected the performance of the solver.

We can still draw some conclusions from these curious results, although these are educated guesses at best and should be viewed as such. First of all, the results from Figure 6.10 are constructed from 72 000 quantum annealing runs and cost multiple minutes of annealing time. Testing whether the difference in performance was related to the launch of the new system would take additional amounts of the finite total quantum annealing time available to us, to the point where other experiments could not be performed. Therefore, we decided to view day to day differences as a possibility and perform most of the remaining experiments on the same day. Secondly, it seems that on a day with deteriorated performance the system experiences more noise, making it advisable to not set the annealing time parameter T too large as the noise would worsen the quantum annealing performance. Finally, it does seem that on both days, an annealing time parameter T of at least 300 microseconds improves the quantum annealing performance. Of course we would recommend to test on a day with a performance similar to or better than that of the first day, but that would be similar to recommending to simply know the optimal solution of the QUBO problem.

Embedding

From the day to day differences perceived, we conclude that quantum annealing performance might be influenced by other variables than we initially expected. Therefore, we also experiment with the amount of qubits within an embedding. Note that these embeddings were found using the sub-optimal minorminer heuristic that is not tailored to the Zephyr (or Pegasus) topology and is made for sparse matrices [16], which the Hamiltonians considered in this report are not. To test the performance of the minorminer heuristic, instance 4 of our dataset was embedded onto the qubits of the *Advantage_system6.4* and the*Advantage2_system1.1* QPUs ten different times. For each embedding the amount of qubits is stored in Table 5.2.

Table 6.1: Number of qubits of embeddings of the 162-variable instance 4 on the *Advantage_system6.4* and the *Advantage2_system1.1* using D-Wave's minorminer heuristic [23] with 10 different random seeds. Per QPU the embeddings with the lowest number of qubits and with the highest number of qubits are highlighted in blue and red, respectively.

Seed	*Advantage_system6.4*	*Advantage2_system1.1*
0	1203	1074
1	1242	996
2	1245	1154
3	1200	1120
4	1348	1181
5	1244	1189
6	1267	1002
7	1309	977
8	1351	1118
9	1356	1005

Within ten times calling the heuristic there is a variation of over 150 qubits, which in this case would equate to roughly one qubit more per variable. All annealing tests presented here and in previous sections were performed on the embeddings that have the lowest amount of qubits. To test if this corresponds to an improved performance, the annealing time T grid search experiments from Figure 6.7, that were performed on the smallest embedding with 977 qubits, have been repeated with the largest embedding found, consisting of 1189 qubits. See Figure 6.11 for the results.

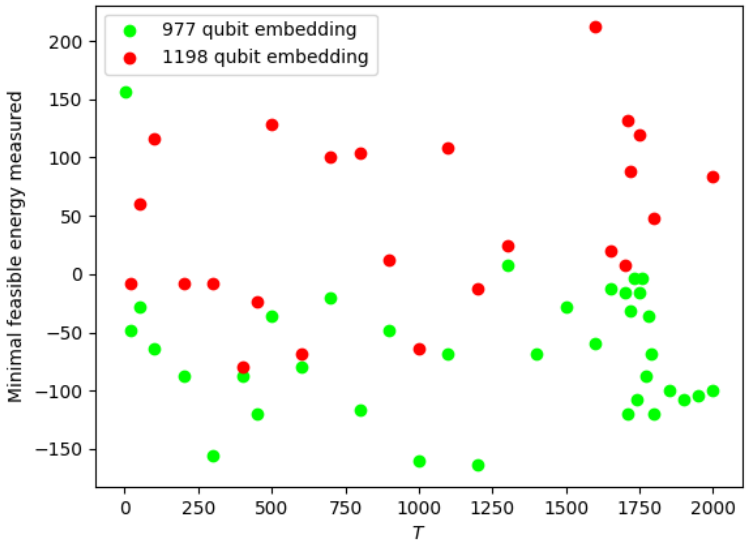

Figure 6.11: Feasible solution with the lowest objective function value taken from a thousand annealing runs per annealing time T value, with results using the largest embedding in red. If no feasible solutions were attained for a value of T, then there is no point plotted. Annealing done on instance 4 with $\lambda = 15$ and default chain strength $cs \approx 105$. The best solution from the larger embedding produces an objective function value of -80 and is attained at $T = 400\,\mu s$.

It is clear that the largest embedding leads to significantly worse performance for all annealing time values, as the largest embedding leads to no feasible samples, or feasible samples with worse performance. Therefore, we will keep using the smallest embedding found for instance 4 or any other Ising model. We did not expect the performance difference to be this large, as the influence of the embedding on the overall performance is typically not mentioned in publications. We consider this result interesting beyond the concrete MOT application considered in this report.

6.4. Chain strength and Lagrangian multiplier

To determine how the penalty parameter λ and the chain strength parameter cs affect the performance of quantum annealing, a thousand quantum annealing runs were performed on all parameter combinations of $\lambda \in \{12, 13, 14, 15, 16, 17\}$ and chain strength $cs \in \{75, 85, 95, \ 105, 115, 125\}$.

All quantum annealing runs were performed using the *Advantage2_system1.2* (see Appendix A) with the 977-qubit embedding and annealing time $T = 600\,\mu s$.

All experiments were repeated on two consecutive days to see whether we would perceive a performance difference presented, similar to the day to day performance difference found during the parameter tuning of the annealing time parameter T (and because of communication with D-Wave explained in Appendix A). Of both days the best performing feasible solutions are in Table 6.2 and the amount of feasible solutions are in Table 6.3. Note that because different values of λ lead to different objective function values, which makes it cumbersome to compare across different Lagrangian multiplier values. Therefore, the results are the objective function value, or Ising model energy, that would be attained using the sample without any penalty parameter, i.e. with $\lambda = 0$.

Table 6.2: Lowest energy attained with a feasible sample taken from thousand runs per λ and cs combination. Results from the first day are displayed in the blue columns and results from the second day in the white columns. If no feasible solution is attained, the cell is left empty.

cs	$\lambda = 12$		$\lambda = 13$		$\lambda = 14$		$\lambda = 15$		$\lambda = 16$		$\lambda = 17$	
75	-124	-164	-148	-168	-140	-164	-100	-96				
85	-148	-168	-176	-152	-120	-92	-112	-140	-48	-164	-112	136
95	-184	-180	-116	-124	-140	-140	-88	-160	-116	-104	-100	-80
~105	-136	-72	-24	-68	-36	-172	-112	-112	-100	-80	-152	-96
115	-136	0	-28	-32	-68	-96	-84	-80	-44	-84	-12	-132
125	224		48	48	16	-60	96	-112	-132	-132	0	-60

On both days $\lambda = 12$ and chain strength 95 returned the best performing feasible solution. There does not seem to be a clear connection between the chain strength and Lagrangian multiplier parameters. Also note, that these results do not indicate a day to day difference in performance. To gain more insight the performance, we also look at the amount of feasible solutions found. See Table 6.3 for the amount of feasible solutions were found out of a thousand runs for all of the grid search parameters.

Table 6.3: Number of feasible samples taken from thousand runs per λ and cs combination. Results from the first day are displayed in the blue columns and results from the second day in the white columns.

cs	$\lambda = 12$		$\lambda = 13$		$\lambda = 14$		$\lambda = 15$		$\lambda = 16$		$\lambda = 17$		Total	
75	19	17	39	23	25	20	5	4	0	0	0	0	88	64
85	8	16	20	51	40	24	36	29	19	26	3	2	126	148
95	31	8	23	17	42	25	39	60	53	32	34	24	222	166
~105	15	3	6	20	16	18	28	108	38	40	54	52	157	241
115	3	3	6	9	14	18	14	59	30	28	27	118	94	235
125	1	0	4	2	1	6	7	63	11	78	27	27	51	176
Total	77	47	98	122	138	111	129	323	151	204	145	223	738	1030

The second day produced almost 300 more feasible solutions, but out of 36 000 solutions that is not considered a large deviation in performance. Although the first day gave fewer feasible solutions, it did give a better

performing minimal feasible solution at chain strength $cs = 95$ and $\lambda = 12$. However, where on the first day these best performing parameters gave 31 feasible solutions, on the second day only 8 out of 1000 samples were feasible solutions. To recap, the second day had better overall performance, but the first day had better performance on the "optimal" parameters found using grid search. In conclusion, the optimal parameter combination was found to be $\lambda = 12$ and $cs = 95$ and no significant day to day performance difference was perceived.

6.5. Annealing schedule

Recall from Section 4.1 that the user may not only set the annealing time T, but also the anneal schedule. On both the *Advantage 1* and the *Advantage 2* it is possible to set up to 20 points when defining the anneal schedule. Fully optimizing these anneal schedule points would require a significant amount of annealing time, while still providing results that probably will not be generalizable beyond this specific dataset. However, there are two interesting features of annealing schedules that we will be testing: adding a pause and adding a quench to the schedule. Both the pause and the quench were introduced in Section 4.1. To recap: a pause is a time during the annealing schedule where energy working on the qubits, and therefore the anneal fraction s, do not change and a quench is terminating (part of) the annealing process as fast as possible.

It is recommended to add a pause to the annealing process at the point where the odds of excitation are the highest, so where $\delta(t) = \delta_{\min}$ [27]. In practice $\delta_{\min}$ and its corresponding t or s values are unknown. Therefore we perform a pause at the quantum critical point (QCP from Section 4.1). The QCP differs per solver, as the $A(s)$ and $B(s)$ functions differ per solver. See Figure 6.12 for these functions and where they cross to find the corresponding QCPs.

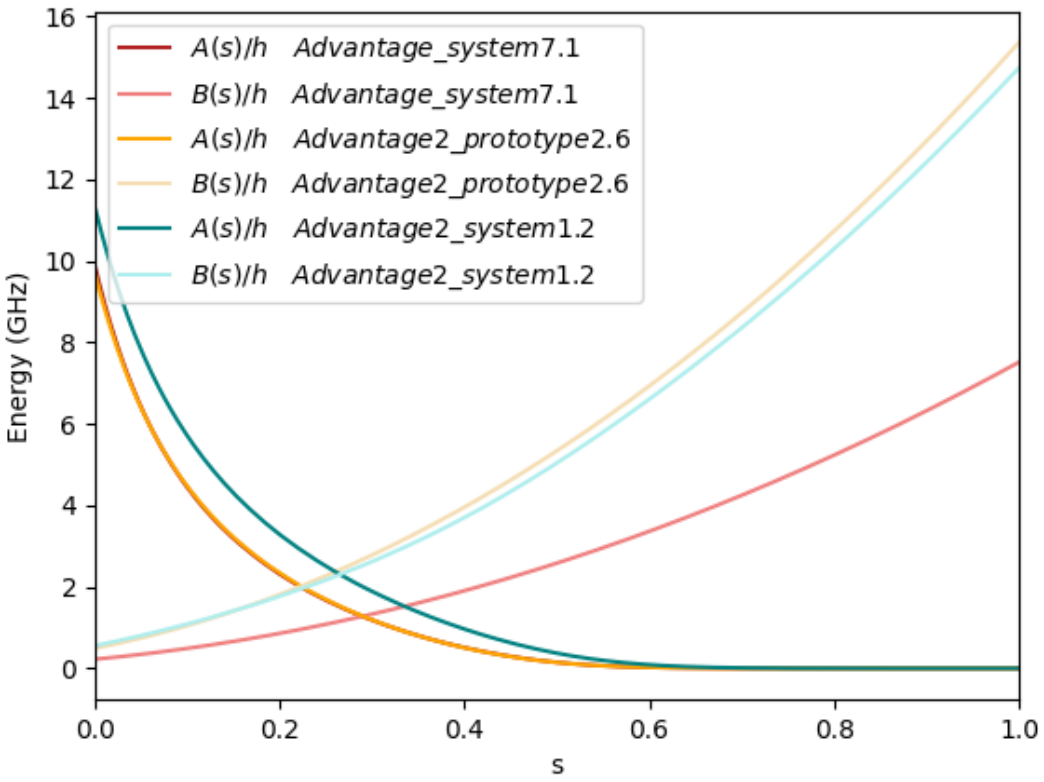

Figure 6.12: $A(s)$ an $B(s)$ functions of the *Advantage_system7.1*, the *Advantage2_prototype2.6* and the *Advantage2_system1.1*. All three solvers have the same $A(s)$ and the *Advantage2_prototype2.6* and *Advantage2_system1.1* have the same $B(s)$ function.

All experiments in this section are performed using the *Advantage 2*, but not all experiments used the same version (1.1, 1.2, or 1.3). These versions have the same annealing functions $A(s)$ and $B(s)$. The prototype does have a different $A(s)$ from the *Advantage 2* and a different $B(s)$ function than the *Advantage 1* and the *Advantage 2*. For the *Advantage 2* the QPS is attained at annealing fraction 0.265.

We tested five anneal schedules that contain a pause. For the first three schedules a pause of 10, 50, and 100 μs, respectively, was performed at $s = 0.265$ and fixed the total annealing time. This produces the following anneal schedule

$$\text{schedule 1} : [(s = 0,\, t = 0),\, (\text{QCP},\, T \cdot \text{QCP}),\, (\text{QCP},\, T \cdot \text{QCP} + \text{pause}),\, (1,\, T)], \tag{6.1}$$

with $T = 600\,\mu s$ and QCP = 0.265. Because the total annealing time was fixed, the speed of annealing was higher outside the pause than it would have been without a pause. To also test with the same annealing speed we also tested on two anneal schedules, where the pause time was added to the total annealing time. Here, a pause of 100 and 500 μs was implemented, giving a total annealing time of 700 and 1100 μs, respectively and corresponds to the following anneal schedule

$$\text{schedule 2} : [(s = 0,\, t = 0),\, (\text{QCP},\, T),\, (\text{QCP},\, T \cdot \text{QCP} + \text{pause}),\, (1,\, T + \text{pause})], \tag{6.2}$$

with again $T = 600\,\mu s$ and QCP = 0.265.

The annealing was performed on instance 4 of the dataset and with the "optimal" chain strength and Lagrangian multiplier values found using grid search in Section 6.4. The tests were performed on the same two consecutive days as the grid search tests from Section 6.4. See Table 6.4 for the energy of the Ising model without penalty parameter of the minimal feasible solution and the number of feasible solutions, again out of a thousand runs, that were found on both days.

Table 6.4: Minimal feasible solution and amount of feasible solutions measured in thousand runs with a pause at QCP, $s = 0.265$. Results from the first day are displayed in the blue columns and results from the second day in the white columns. If no feasible solution is attained, the cell is left empty.

Pause μs	Total μs	Min. feas. E		# feas.	
10	600	-144	-176	4	10
50	600	-68	-92	9	9
100	600	-100	-8	4	3
100	700	-24	-16	4	4
500	1100		48	0	2

The highest performance was attained with a pause of 10 μs which may be negligible on a total annealing time of 600 μs. Also, the addition of a pause did not produce more feasible solutions or a better feasible solution than found during annealing time T parameter testing (energy of -184, see Figure 6.7), or during Lagrangian multiplier λ and chain strength cs parameter grid search (energy of -184, see Table 6.2).

As for adding a quench to the annealing schedule, we experimented with ten different anneal schedules containing a quench. The quench starts at anneal fraction $s = s_{\text{qnch}}$ where we tested anneal fractions $s_{\text{qnch}} \in \{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8\}$. The anneal schedule then becomes the following:

$$\text{schedule 3} : [(s = 0,\, t = 0),\, (s_{\text{qnch}},\, T \cdot s_{\text{qnch}}),\, (1,\, T \cdot s_{\text{qnch}} + 0.5(1 - s_{\text{qnch}}))], \tag{6.3}$$

where we set $T = 600\,\mu s$. The total annealing time then becomes $600 \cdot s_{\text{qnch}}\,\mu s + 0.5(1 - s_{\text{qnch}})\,\mu s$, which is below 600 μs. In case this reduction of total annealing time affects the quantum annealing performance, we also used the following schedule

$$\text{schedule 4} : [(s = 0,\, t = 0),\, (s_{\text{qnch}},\, T),\, (1,\, T + 0.5(1 - s_{\text{qnch}}))], \tag{6.4}$$

with $T = 600\,\mu s$ and $s_{\text{qnch}} \in \{0.5, 0.65, 0.8\}$. The experiments were performed on dataset instance 4 of a MOT problem with the best performing parameters found in Section 6.4 and were performed on the same days

as the grid search and pause experiments. See Table 6.5 for the energy of the Ising model without penalty parameter of the minimal feasible solution and the number of feasible solutions, again out of a thousand runs, that were found on both days.

Table 6.5: Minimal feasible solution and amount of feasible solutions measured in thousand runs for different quench and annealing time T values. Results from the first day are displayed in the blue columns and results from the second day in the white columns. If no feasible solution is attained, the cell is left empty.

Quench s	Total s	Min. feas. E		# feas.	
0.5	300.25	-52	-108	8	9
0.5	600.25		-200	0	18
0.55	330.225	-104	-88	5	9
0.6	360.2	-128	-116	6	10
0.65	390.175	-116	-84	5	8
0.65	600.175	-56	-104	7	13
0.7	420.15	-92	-156	21	6
0.75	450.125	-164	-136	9	8
0.8	480.1	-108	28	9	4
0.8	600.1	-108	-68	18	5

The performance is similar to that of Section 6.4 with chain strength 95 and Lagrangian multiplier equal to 12. It is not clear which day had better performance. What is most interesting, is that an objective function value of -200 was found, which improves the previous best of -184 (see Table 6.2). It would certainly be of interest to further test different anneal schedules containing a quench, as the best performance (on the second day) was attained at the lowest value for s_{qnch}.

6.6. Reverse annealing

Reverse annealing, as introduced in Section 4.1, starts the quantum system with anneal fraction $s = 1$, meaning the initial energy working on the state is actually the problem Hamiltonian instead of the initial Hamiltonian. Reverse annealing comes with two additional inputs: the initial state and the reversal distance.

Definition 6.1. The **reversal distance** in a reverse annealing schedule is the distance between 1, the highest anneal fraction, and s_r, the lowest anneal fraction obtained, i.e.

$$\text{reversal distance} = 1 - \min\{s, s \in \text{anneal schedule}\} := 1 - s_r.$$

For a standard quantum annealing process, with or without a specified schedule, the quantum system is initialized to be in state $|\phi_0\rangle_{\hat{H}_I} = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ with $\hat{H}_I$ the initial Hamiltonian (see Equation 4.1). This initialization is fixed and therefore the same for any problem Hamiltonian the user may input. However, for reverse annealing one can specify this initial state to be any classical state. This initial state may be a feasible solution to the Ising model, in which case the reverse annealing process has higher odds of the quantum state transforming into other feasible states, which could result in better performance than the initial state.

In this section we will also use a feasible state as initial state and perform reverse annealing on our Ising model (corresponding to dataset instance 4). In these experiments we are interested in the performance compared to the previous annealing processes performed on instance 4, as well as the performance compared to that of a paper by D-Wave that demonstrates the capabilities of reverse annealing [28].

In 2017 reverse annealing was first demonstrated on the D-Wave *2000Q*. The Ising model that was used for the demonstration contains clusters of ground states, where, "within a cluster, ground states are separated by tall, thin energy barriers that make quantum annealing valuable even at a local level" as stated by King et al. [3].

King et al. fixed one of the ground states as the initial state and performed reverse quantum annealing with different reversal distance values. The resulting samples were classified as the same state, which is the initial state, as a new ground state, or as not a ground state. For new ground states King et al. also considered the Hamming distance between the same/original and the new ground states.

Definition 6.2. The **Hamming distance** between two samples/states/vectors $x, y \in \mathbb{Z}^n$ is equal to the number of indices where x and y differ

$$d(x, y) := \sum_{i=1}^{n} \mathbb{1}_{x_i \neq y_i}.$$

King et al. produced the results depicted in Figures 6.13 and 6.14 and established the "Goldilocks" region, which is the range of the reversal distance that produced the highest probability of obtaining a new ground state.

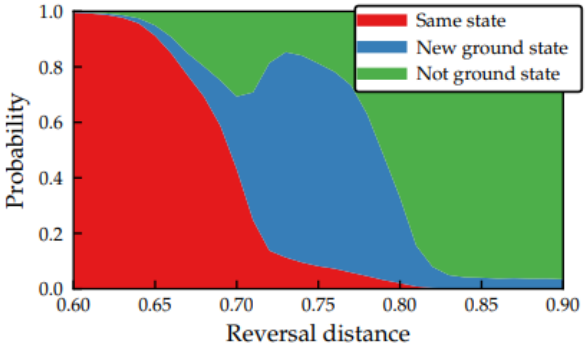

Figure 6.13: Figure from [3] with original caption: "Transition types as a function of reversal distance. If the reversal distance is too low, the algorithm remains in the starting state; if it is too high, the algorithm fails to exploit knowledge of the initial state. "

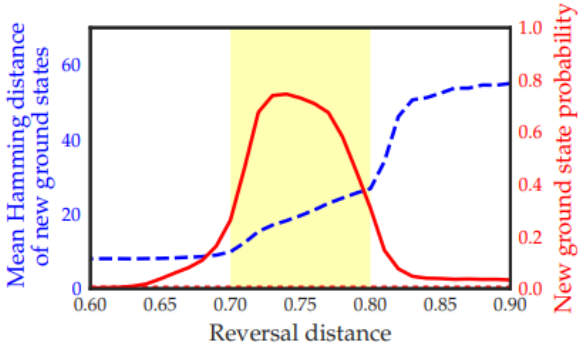

Figure 6.14: Figure from [3] with original caption: "Probability of reaching a new ground state (red) and mean Hamming distance of new ground states from initial states (blue) as functions of reversal distance. Yellow shading is used to indicate the "Goldilocks" region in which we are reversing far enough to escape the initial ground state but not so far that we lose the ability to exploit knowledge of the initial state; the probability of arriving at a new ground state peaks in this region. The mean distance of new ground states from their starting states increases with reversal distance."

MOT problems do not have clusters of ground states and most likely have a unique ground state (depending on the cost calculation). Therefore, we do not consider the Hamming distance during testing and focus on whether reverse annealing can produce better feasible states for MOT problems. To test this we performed reverse annealing with an annealing time of $T = 600\,\mu s$ on dataset instance 4 with reversal distance $\in \{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9\}$ on four parameter settings:

1. Default chain strength ($cs \approx 105$) and the Lagrangian multiplier value found using simulated annealing ($\lambda = 15$). Reinitialization the initial state to be the same for each run.
2. Default chain strength ($cs \approx 105$) and the Lagrangian multiplier value found using simulated annealing ($\lambda = 15$). The output sample of a run is used to initialize the following run.
3. Optimal chain strength ($cs = 95$) and Lagrangian multiplier ($\lambda = 12$). Reinitialization the initial state to be the same for each run.
4. Optimal chain strength ($cs = 95$) and Lagrangian multiplier ($\lambda = 12$). The output sample of a run is used to initialize the following run.

For each combination of reversal distance and parameter setting a thousand runs were performed and samples were classified as infeasible, the same as the initial feasible state, feasible with worse objective function value than the initial feasible state, or as feasible with better objective function value than the initial feasible state (which would be the desired outcome). The initial state is set to be a feasible state with an objective function value of -176 (slightly worse than the highest performing quantum annealing results that were obtained previously) and is obtained by performing ten runs of simulated annealing.

The results are visualized in the same way as by King et al. in Figure 6.13 and are depicted in Figures 6.15 (default parameters, with reinitialization), 6.16 (default parameters, without reinitialization), 6.17 (optimized parameters, with reinitialization), and 6.18 (optimized parameters, without reinitialization).

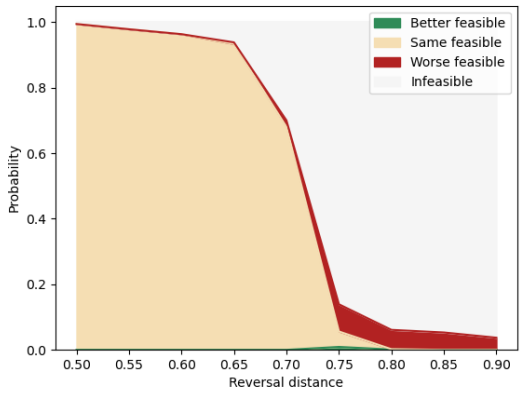

Figure 6.15: Default chain strength of ~ 105, $\lambda = 15$ and annealing time $T = 600\,\mu s$. With reinitialization of initial state. Thousand reverse annealing runs per reversal distance.

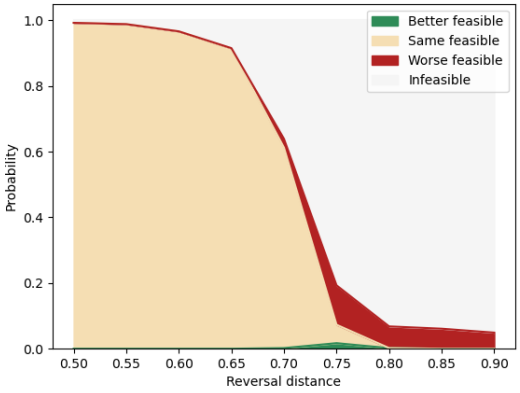

Figure 6.16: Default chain strength of ~ 105, $\lambda = 15$ and annealing time $T = 600\,\mu s$. Without reinitialization of initial state, i.e. the result of the previous run is the initial state of the new run. Thousand reverse annealing runs per reversal distance.

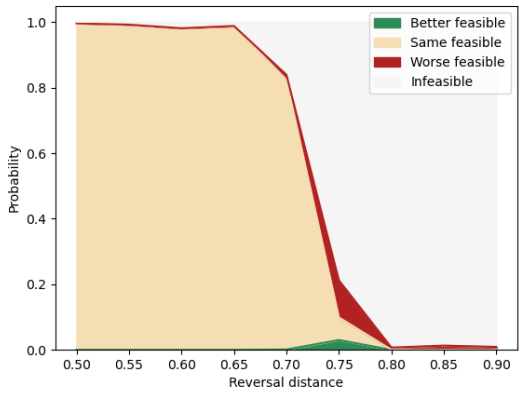

Figure 6.17: Optimized chain strength of 95, $\lambda = 12$ and annealing time $T = 600\,\mu s$. With reinitialization of initial state. Thousand reverse annealing runs per reversal distance.

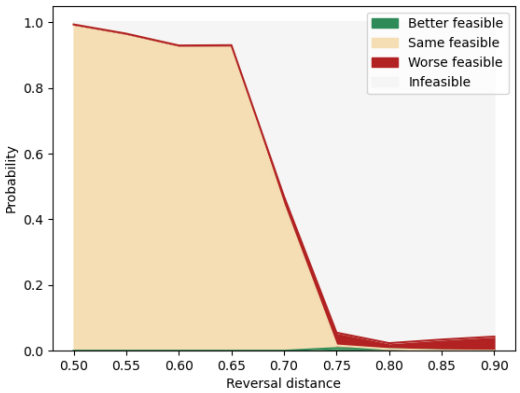

Figure 6.18: Optimized chain strength of 95, $\lambda = 12$ and annealing time $T = 600\,\mu s$. Without reinitialization of initial state, i.e. the result of the previous run is the initial state of the new run. Thousand reverse annealing runs per reversal distance.

If the reversal distance is below 0.65, then there is a 90% chance the system measured will be the same as its initial state. If the reversal distance is higher, then the energy working on the system changes more, which leads to new states of the quantum system. Most of these not-initial states are infeasible, which is not surprising as the majority of the 2^{162} possible states are not feasible.

For a reversal distance of 0.7, 0.75, or 0.8 the reverse annealing produced multiple samples that were feasible and with a better/lower objective function value. For a reversal distance of 0.85 exactly one sample with increased performance was observer. For the other reversal distances there were no better performing feasible solutions found. In Table 6.6 the number of better performing feasible solutions is stored together with the best feasible solution found.

Table 6.6: Minimal feasible solution and amount of feasible solutions measured in reverse annealing runs with different schedules and reversal distances. Initial state produces a objective function value of -176. If the best feasible solution was worse than the initial, the result is highlighted in red. Best results are highlighted in green.

Parameters	Reinitialization	Min. feas. E				# feas. improved			
		Reversal distance				Reversal distance			
		0.7	0.75	0.8	0.85	0.7	0.75	0.8	0.85
Default	True	-176	-208	-216	-60	0	9	1	0
Default	False	-180	-208	-212	-116	2	17	2	0
Optimal	True	-180	-216	-88	-96	1	30	0	0
Optimal	False	-176	-208	-212	-180	0	8	5	1

Reverse annealing, with a reversal distance of 0.75, produces the most better feasible states. These results are comparable to the “Goldilocks” region from [28], which is quite remarkable considering that the Goldilocks region was based on results from 8 years ago. Therefore, this “Goldilocks” region seems to be robust with respect to the solver, including denoising methods, topology, annealing time and overall newer technology, and the Ising models used. Reverse annealing produces significantly better results than with the standard anneal path. Note that the reverse annealing results, and why it performs better than standard quantum annealing, can not be explained with the theory on quantum annealing from Section 2 or with simulated annealing.

0111

Conclusions

After establishing a base performance on our MOT dataset in Section 5, the influence of annealing time T, Lagrangian multiplier λ, the chain strength cs, and the anneal schedule were further analysed in Section 6. First we provide some conclusions that are related to the dataset that was tested on, including the influence of the T, λ, and cs as well as the anneal schedule. Then we move to general Ising models for quantum annealing by providing a full pipeline for future quantum annealer users.

7.1. Dataset specific conclusions

For instance 4 of the MOT dataset 2010 simulated annealing runs were performed and a total of 250 000 quantum annealing runs, of which 246 000 on the *Advantage 2* annealer. Of the quantum annealing runs the best result, i.e. the feasible solution with the lowest Ising model value, was obtained for reverse annealing, and was equal to -216. For a standard annealing path the best result was equal to -200, which was obtained when adding a quench at $s = 0.5$ with total annealing time of $T = 600.25\,\mu s$.

Parameter engineering of the annealing time was effective in the sense that low annealing times such as the default $20\,\mu s$ are too small for the instances of our dataset with over a 100 variables. The parameter engineering for the chain strength and Lagrangian multiplier parameters did not lead to a significant increase in performance. What was clear, however, is that the performance drops when increasing one of the parameters while decreasing the other, as both influence the embedded Ising model and therefore increasing one can diminish the effect of the other. In other words, if the Lagrangian multiplier is increased while the chain strength is decreased, then the effect of the chain strength diminishes during the annealing, and vice versa.

Adding a pause to the annealing schedule seemed promising from a theoretical standpoint, as it would lower the chance of excitation during the pause. The pause was performed at the QCP, since it was unknown at which point in the annealing the gap would be the smallest. Determining $\delta_{\min}$ could give more insight on where to add a pause to the schedule. However, it is not advised to try to find the $\delta_{\min}$ for the larger Ising models from this report or any other Ising model with over a hundred variables. Therefore, we can only conclude that adding a pause at the QCP did not improve the annealing performance of the Ising model considered.

Adding a quench did seem to improve performance in some cases. However, as was the case with the parameter engineering, the performance tended to differ per day.

Where most experiments differed by day, the performance of reverse annealing was similar to that of a different set of experiments performed years ago and on a two generations older QPU. The four cases that were tested consisted of using the default or the earlier found optimal parameters, and reinitializing the initial state of the system or not. The four cases combined with a reversal distance of 0.75, which is the middle of the "Goldilocks region", defined by D-Wave [28], all returned feasible solutions with lower objective function value than any of the results from the standard annealing paths. Reverse annealing does require an initial state, however, this was easy and quick to find using simulated annealing.

7.2. Full quantum annealing pipeline

The final conclusions will be formulated as advise for future quantum annealing users that have limited annealing time and do not have access to the underlying solution of the problem, as this is most likely the situation for quantum annealing users (that do not research quantum annealing itself).

As a full pipeline we recommend the following steps:

1. Set the chain strength to default and either find the Lagrangian multiplier via simulated annealing, or set it to the same value as Ising models of similar size.
2. If possible, employ the heuristic to find the embedding with varying random seeds. Set the embedding with the least amount of qubits as the embedding used during the quantum annealing process. Allot up to one full workday for this step, as it will take a long time to find (multiple) embeddings.
3. Use simulated annealing to get a feasible solution. If this is not possible due to time constraints, then choose any classical state, e.g., set every decision variable to 1.
4. Use reverse annealing with a reversal distance of 0.75 and set the initial state to the state from step 3.
5. Do up to a thousand runs of reverse annealing using one of the following schedules:

$$\text{Schedule without quench} = [(1,0),(0.25,300),(1,600)], \tag{7.1}$$

$$\text{Schedule with quench} = [(1,0),(0.25,450),(0.5,600),(1,600.25)]. \tag{7.2}$$

6. Find the best performing (feasible) sample from step 5. If one wishes to further improve upon the performance, return to step 4 using the best performing sample as initial state.

1000

Discussion and Future Work

The results presented in this report are often based on educated guesses, as the amount of literature applicable to quantum annealing on MOT problems was scarce. These educated guesses were not always correct and therefore we discuss some of the oversights made in this thesis as well as discuss some of the challenges present during the experiments. Finally, we propose some research that could be done in the future to improve the understanding of quantum annealing.

One oversight was made with regards to the chain strength, as there are already multiple ways of setting the chain strength, including the incorporation of the full embedding. D-Wave provides different ways of setting the chain strength, but this was discovered too late into the quantum annealing testing that it was not incorporated.

Another oversight was made with regards to the anneal fraction at which the pause was performed. Where we assumed the QCP to be a suitable anneal fraction to add a pause to the schedule, it may be that another anneal fraction would lead to better performance. For example, Marshall et al. observed an improvement in performance by adding a pause [27]. However, the pauses were added at higher anneal fractions. We first assumed this to be an old result or a result that was specific to the Ising model considered by Marshall et al., but after experimenting with reverse annealing and perceiving the range of the reversal distance that leads to better performance, we no longer assume this to be the case. Therefore, we recommend to test the addition of a pause to the anneal schedule again, but by adding the pause at other anneal fractions than the QCP.

Additionally, D-Wave provides many more ways of tailoring the quantum annealing experiments, such as setting up to twenty points in the anneal schedule, while we only tested on anneal schedules with up to four points. D-Wave also provides a feature named “fast annealing” which contains smaller annealing times T and is more noise resistant. Finally, D-Wave even provides defining an anneal offset, which essentially enables the quantum annealer user to fine-tune the annealing process to specific qubits within the embedding.

Where initially the focus of this report was on optimizing the annealing time T, Lagrangian multiplier λ, and the chain strength cs parameters, we found that the embedding also significantly affects the results. Additionally, the day-to-day differences in quantum annealing performance affected the results as well, making it unclear whether a change in performance was due to parameters, unknown variables that could differ per day, or pure coincidence.

The substantial day-to-day quantum annealing performance differences may be related to the release of the *Advantage 2* and may already be resolved. This is unknown, however, and would require additional testing. We speculate that knowing the temperature of the QPU at the time of testing could help with the estimation of the QPU performance, if there is a relation between the temperature of the QPU and the quantum annealing performance. However, this would be more so related to the necessity of better understanding noisy quantum systems than quantum annealing itself.

To further research quantum annealing we propose the following improvements:

- Improving the embedding heuristic and tailoring it to the topology of the QPU.

- Researching the effect of the sparsity of matrix **Q** on the quantum annealing performance.
- Researching how the quantum annealing performance scales for larger problems and if it scales with the amount of qubits or the amount and shape of chains.
- Doing parameter optimization for reverse annealing.
- Researching whether the parameter optimization is generalizable to all Ising models of the same sparsity and/or size.

It is safe to say that a lot of the specifics of quantum annealing remain to be discovered. We advise to experiment on the newest quantum annealers, and general Ising models to further the research on quantum annealing.

Bibliography

[1] Laura Leal-Taixé et al. *MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking*. 2015. arXiv: 1504.01942.

[2] S. van Benthem. *Track and Truth Correlation in Military Simulations*. 2024.

[3] James King et al. *Quantum Annealing amid Local Ruggedness and Global Frustration*. 2017. arXiv: 1701.04579.

[4] *Learning to program the D-Wave One*. Accessed on 12/06/2025. URL: https://web.archive.org/web/20110723043401/http://dwave.wordpress.com/2011/05/11/learning-to-program-the-d-wave-one/.

[5] R. Harris et al. "Experimental demonstration of a robust and scalable flux qubit". In: *Physical Review B* 81.13 (Apr. 2010). ISSN: 1550-235X. DOI: 10.1103/physrevb.81.134510.

[6] Jan-Nico Zaech et al. *Adiabatic Quantum Computing for Multi Object Tracking*. 2022. arXiv: 2202.08837.

[7] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[8] Wikipedia. *Definition of the ground state*. Accessed on 04/07/2025. URL: https://en.wikipedia.org/wiki/Ground_state.

[9] Edward Farhi et al. *Quantum Computation by Adiabatic Evolution*. 2000. arXiv: quant-ph/0001106.

[10] M. Born and V. Fock. "Beweis des Adiabatensatzes". In: *Zeitschrift für Physik* 51 (1928), pp. 165–180. DOI: 10.1007/BF01343193.

[11] Albert Messiah. *Quantum Mechanics*. John Wiley and Sons, 1958.

[12] Joseph E. Avron and Alexander Elgart. "Adiabatic Theorem without a Gap Condition". In: *Communications in Mathematical Physics* 203.2 (June 1999), pp. 445–463. DOI: 10.1007/s002200050620.

[13] Fred Glover, Gary Kochenberger, and Yu Du. *A Tutorial on Formulating and Using QUBO Models*. 2019. arXiv: 1811.11538.

[14] Andrew Lucas. "Ising formulations of many NP problems". In: *Frontiers in Physics* Volume 2 - 2014 (2014). DOI: 10.3389/fphy.2014.00005.

[15] *Reverse annealing tutorial notebook*. Accessed on 15/06/2025. URL: https://github.com/dwave-examples/reverse-annealing-notebook/blob/master/01-reverse-annealing.ipynb.

[16] Jun Cai, William G. Macready, and Aidan Roy. *A practical heuristic for finding graph minors*. 2014. arXiv: 1406.2741.

[17] *Anneal schedule tutorial notebook*. Accessed on 15/06/2025. URL: https://github.com/dwave-examples/anneal-schedule-notebook/blob/master/01-anneal-schedule.ipynb.

[18] *QPU solver datasheet from the D-Wave website*. Accessed on 13/02/2025. URL: https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html.

[19] *Annealing Implementation and Controls from the D-Wave website*. Accessed on 15/02/2025. URL: https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html.

[20] *D-Wave topologies*. Accessed on 15/06/2025. URL: https://docs.dwavequantum.com/en/latest/quantum_research/topologies.html.

[21] Wikipedia. *Definition of a complete bipartite graph*. Accessed on 04/07/2025. URL: https://en.wikipedia.org/wiki/Complete_bipartite_graph.

[22] D-Wave whitepaper. *Programming the D-Wave QPU: Setting the Chain Strength*. 2020.

[23] *Github of minorminer*. Accessed on 18/05/2025. URL: https://github.com/dwavesystems/minorminer.

[24] *D-Wave's default chain strength method.* Accessed on 28/06/2025. URL: https://docs.dwavequantum.com/en/latest/ocean/api_ref_system/generated/dwave.embedding.chain_strength.uniform_torque_compensation.html#dwave.embedding.chain_strength.uniform_torque_compensation.

[25] Amin Barzegar et al. *Optimal schedules for annealing algorithms.* 2024. arXiv: 2402.14717.

[26] *Github repository containing all quantum annealing results.* URL: https://github.com/MirteSudoku/Quantum-Annealing.git.

[27] Jeffrey Marshall et al. "Power of Pausing: Advancing Understanding of Thermalization in Experimental Quantum Annealers". In: *Physical Review Applied* 11.4 (Apr. 2019). DOI: 10.1103/physrevapplied.11.044083.

[28] D-Wave whitepaper. *Reverse Quantum Annealing for Local Refinement of Solutions.* 2011.

Appendix A

Timeline of D-Wave solver system updates and quantum annealing experiments performed.

22/04/2025 Access to D-Wave solvers for this thesis was provided by TNO.

14/05/2025 Obtained preliminary results on the *Advantage_system7.1* and the *Advantage2_prototype2.6* solvers.

20/05/2025 *Advantage2_system1.1* was released to the public.

20/05/2025 *Advantage_system7.1* was decommissioned but still available with decreased performance.

30/05/2025 First day of annealing time parameter experiments on the *Advantage2_system1.1*.

31/05/2025 Second day of annealing time parameter experiments on the *Advantage2_system1.1*.

17/06/2025 *Advantage2_system1.2* replaces the *Advantage2_system1.1* solver.

20/06/2025 First day of grid search (of λ and cs), pause and quench testing on the *Advantage2_system1.2* solver.

21/06/2025 *Advantage2_system1.3* replaces *Advantage2_system1.2*.

21/06/2025 Second day of grid search (of λ and cs), pause and quench testing on the *Advantage2_system1.3* solver.

24/06/2025 Reverse annealing testing on the *Advantage2_system1.3* solver.

25/06/2025 *Advantage2_prototype2.6* is decommissioned.

Appendix B

Comparison of annealing time T of 400 and 2000 μs performance on the *Advantage_system7.1* solver.

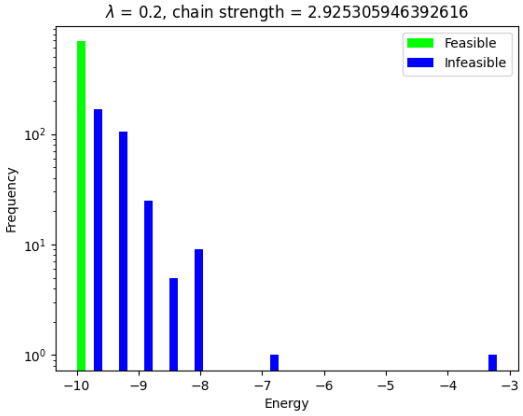

Figure 8.1: Instance 1, 400 μs on the *Advantage_system7.1* solver.

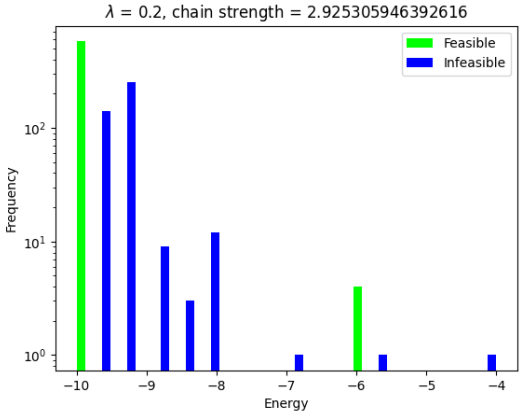

Figure 8.2: Instance 1, 2000 μs on the *Advantage_system7.1* solver.

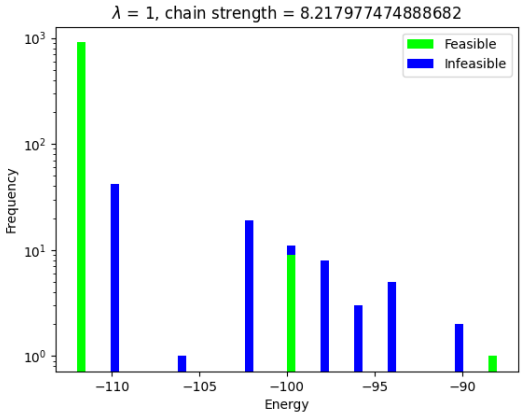

Figure 8.3: Instance 2, 400 μs on the *Advantage_system7.1* solver.

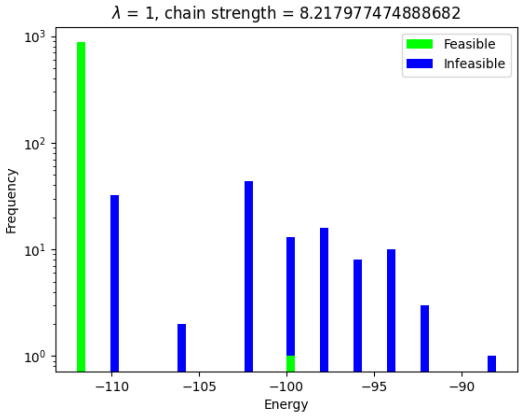

Figure 8.4: Instance 2, 2000 μs on the *Advantage_system7.1* solver.

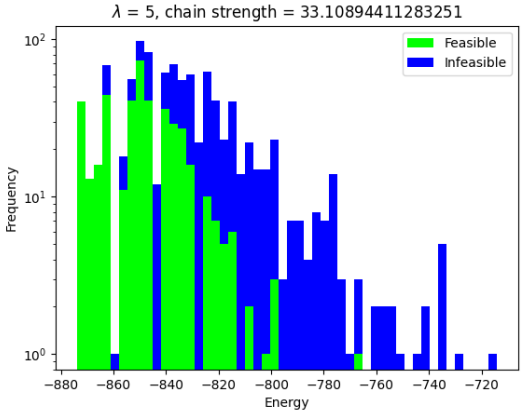

Figure 8.5: Instance 3, 400 μs on the *Advantage_system7.1* solver.

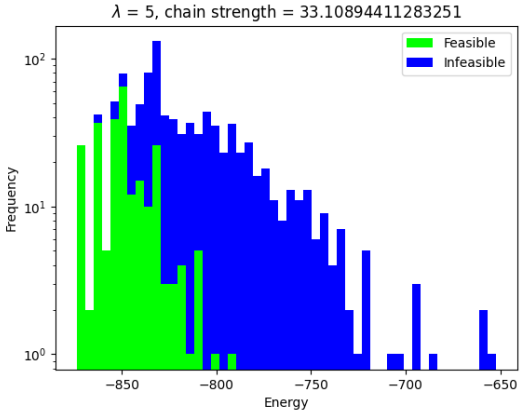

Figure 8.6: Instance 3, 2000 μs on the *Advantage_system7.1* solver.

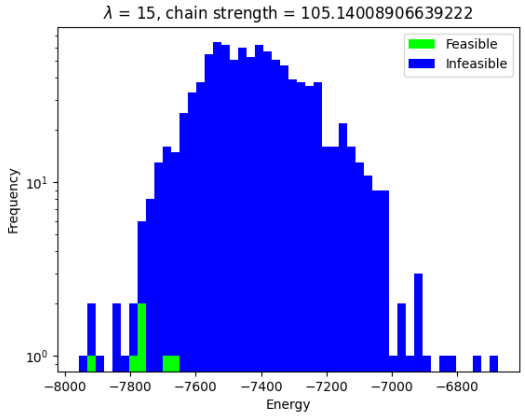

Figure 8.7: Instance 4, 400 μs on the *Advantage_system7.1* solver.

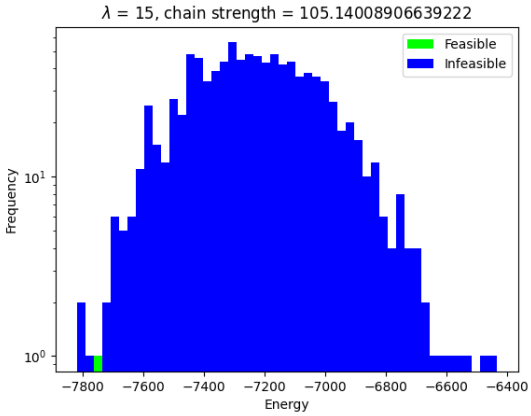

Figure 8.8: Instance 4, 2000 μs on the *Advantage_system7.1* solver.

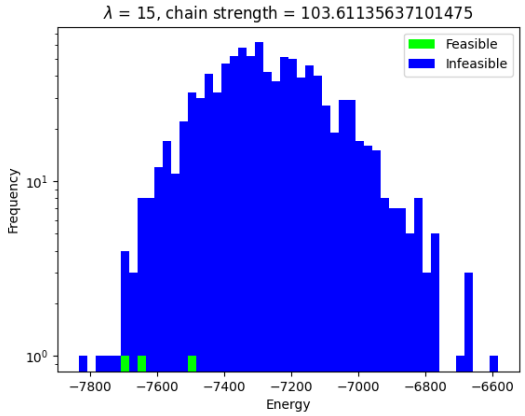

Figure 8.9: Instance 5, 400 μs on the *Advantage_system7.1* solver.

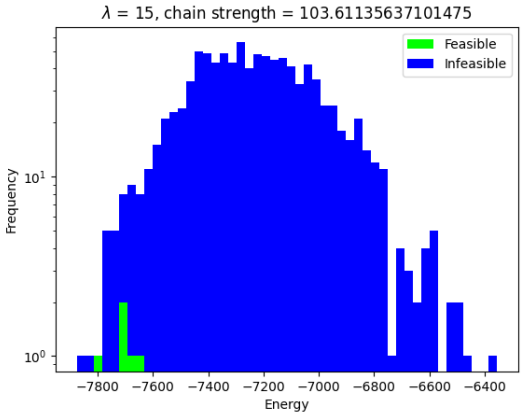

Figure 8.10: Instance 5, 2000 μs on the *Advantage_system7.1* solver.

Comparison of annealing time T of 400 and 2000 μs performance on the *Advantage2_prototype2.6* solver.

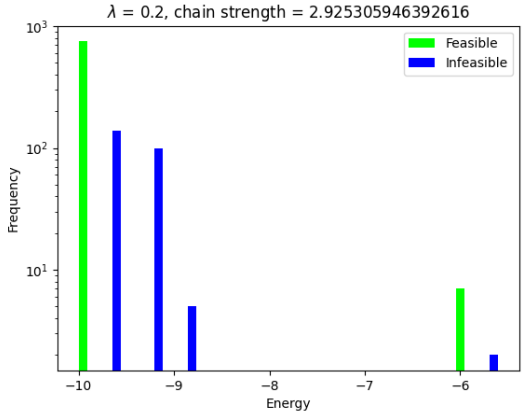

Figure 8.11: Instance 1, 400 μs on the *Advantage2_prototype2.6* solver.

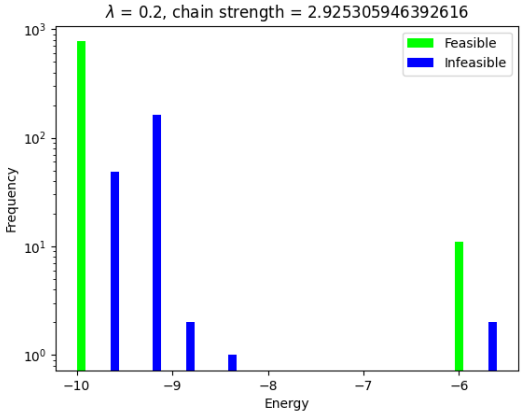

Figure 8.12: Instance 1, 2000 μs on the *Advantage2_prototype2.6* solver.

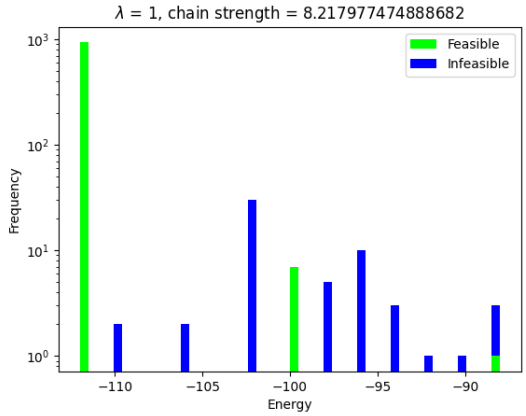

Figure 8.13: Instance 2, 400 μs on the *Advantage2_prototype2.6* solver.

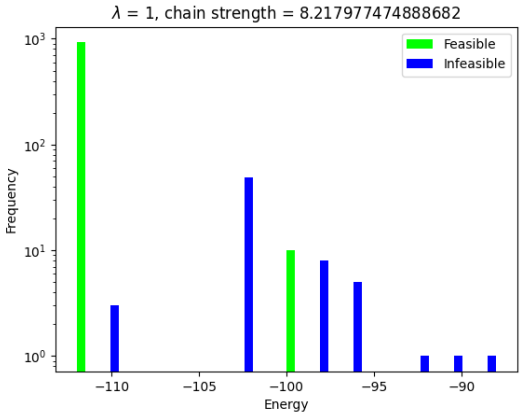

Figure 8.14: Instance 2, 2000 μs on the *Advantage2_prototype2.6* solver.

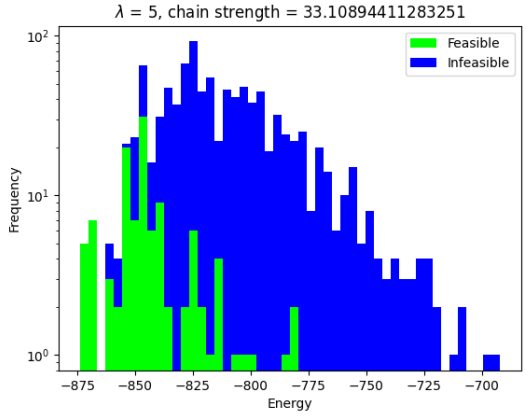

Figure 8.15: Instance 3, 400 μs on the *Advantage2_prototype2.6* solver.

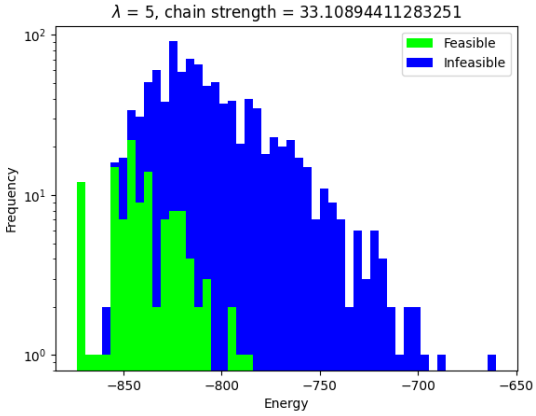

Figure 8.16: Instance 3, 2000 μs on the *Advantage2_prototype2.6* solver.

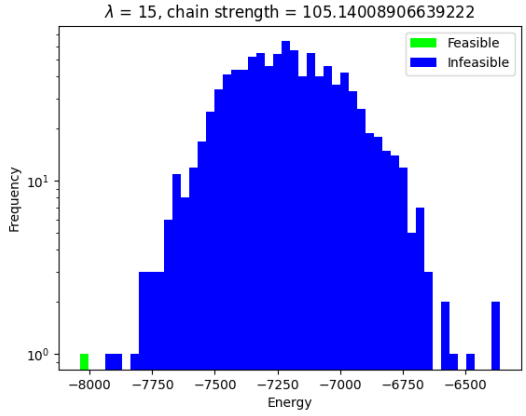

Figure 8.17: Instance 4, 400 μs on the *Advantage2_prototype2.6* solver.

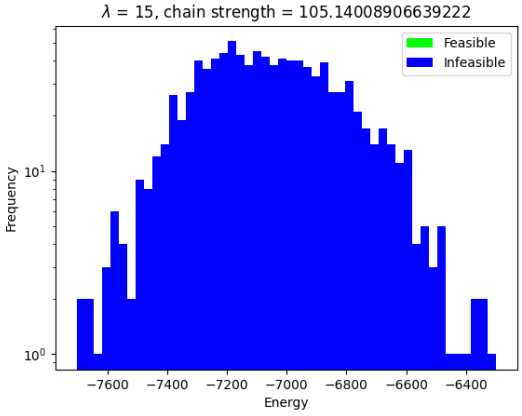

Figure 8.18: Instance 4, 2000 μs on the *Advantage2_prototype2.6* solver.