Quantum annealing for music arrangement

Lucas Kirby

Department of Physics, University of Durham

Supervised by

Prof Robert Potvliege & Dr Omer Rathore

23 April 2025

Abstract

Music arrangement is usually a complex and time-consuming process; this paper aims to provide an automatic method by which to arrange music via a quantum computing technique called quantum annealing. By splitting a score into a set of phrases, these phrases can form a quadratic unconstrained binary optimisation (QUBO), a function which the quantum computer aims to minimise by choosing the values of discrete variables. At the end of the optimisation process, the resulting chosen values describe the final arrangement, which can then be interpreted into sheet music.

Contents

1 Introduction

The field of quantum computing has its foundations as early as the 1980s, with the suggestion that hardware following the laws of quantum mechanics could be faster and more powerful than its classical counterpart [1]. Quantum computers, which leverage quantum mechanical phenomena to perform calculations, store information as qubits ("quantum bits") which can exploit effects such as superposition and entanglement to increase computing power. As quantum computers developed, along with them came the possibility to solve complex problems usually intractable to even the most powerful classical computers, an advantage dubbed "quantum supremacy" [2]. This is not to say that quantum computers can solve problems that are impossible for classical computers, but that they can solve them faster. A collaboration between Google AI Quantum and NASA became the first to claim quantum supremacy in 2019 [3], although this claim has since been refuted [4, 5]. More recently, D-Wave Quantum Systems Inc. claimed to have demonstrated quantum supremacy simulating quantum critical dynamics, performing magnetic materials simulations in minutes that would take nearly one million years using classical supercomputers [6]. However, this claim too was quickly challenged [7, 8]. Skepticism about quantum computers has existed since their conception [9] and is still ongoing [10], but this does not prevent many from being optimistic about future applications.

Since its inception, two methods of quantum computing have been developed: gate-based computing and adiabatic quantum computing. Gate-based quantum computers use quantum gates to manipulate qubits, analogous to classical logic gates for conventional circuits, where gates can be represented as operators acting on qubit states. This kind of computer is versatile and well-suited to solving general problems, and is being actively developed by several technology companies as the next step in general computing [11, 12]. Adiabatic quantum computers, on the other hand, rely on the application of the adiabatic theorem to slowly evolve a quantum system. This is often used to find the global minimum of a given function which would be almost impossible to solve analytically. Adiabatic quantum computers have been used to find solutions to a variety of optimisation problems, such as protein folding [13], financial portfolios [14], and traffic flow [15], although only within the limits of classical optimisation algorithms.

The combination of computing and music is a relatively new field, with the involvement of quantum computers even more so. Music is often seen as a very human endeavour, where only skilled musicians could compose and perform such sequences of sound that would be considered art. The idea that computers could produce music arose as early as the 19th century, with the conception of the first general-purpose computer, the Analytical Engine[1]. It was noted that, given "the fundamental relations of pitched sounds in the science of harmony and of musical composition... the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent" [16]. The first computer composition would not be produced until

[1]The computer was never actually built, but laid the foundation for many modern computing concepts.

nearly a century later [17], and since then various classical approaches to computer music have been taken, including Markov chains, evolutionary models, and neural networks [18]. Quantum computing is the latest iteration of computer music, opening up new and unique possibilities.

Since its emergence into the computer music scene, quantum computers have been used to write melodies [19], develop harmonies [20], create vocal synthesisers [21], and create intelligent musical systems via quantum natural language processing (QNLP) [22], using a mixture of gate-based and adiabatic methods. It is emphasised that, whilst these techniques could just as well be implemented on classical computers, the motivation is "the music technology community should be quantum-ready for when quantum computing hardware becomes more sophisticated" [21]. However, all these efforts have been directed at music *composition*, that is, the generation of entirely new sequences and sounds, rather than music *arrangement*, the adaptation of previously-composed pieces. Indeed, at time of writing, the application of quantum computing to the problem of music arrangement has never been explored before.

In this study, we propose that music arrangement can be formulated as an optimisation problem to be solved using a heuristic application of adiabatic quantum computing called quantum annealing. The structure of the study is as follows: we first introduce the relevant theory for both quantum annealing as a computational tool, and music arrangement as a suitable problem to solve; we then propose an original framework for combining the two, and apply this method to two musical scores; finally, we consider conclusions and possible future work.

2 Quantum annealing

As mentioned, quantum annealing falls under the category of adiabatic quantum computing (AQC). AQC is a computing technique that relies on the adiabatic theorem [23].

Theorem (Adiabatic theorem)**.** *A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.*

A linear evolution of the Hamiltonian of a system can be expressed as

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_p \tag{1}$$

where the system is evolving from an initial Hamiltonian H_0 to a final Hamiltonian H_p over a time interval T. According to the adiabatic theorem, if the system starts in the nth eigenstate of H_0, then, given that the evolution is slow enough and that there is some energy difference between eigenstates (i.e. non-degeneracy), it will remain in the nth instantaneous eigenstate of $H(t)$ throughout evolution and end in the nth state of H_p. How "slow" the evolution needs to be is determined by the minimum energy gap of the instantaneous Hamiltonians; an approximate

adiabaticity criterion can be given by

$$\frac{\hbar \langle \psi_m | \dot{H} | \psi_n \rangle}{(E_n - E_m)^2} \ll 1 \tag{2}$$

where we have used the largest matrix element between a pair of adjacent eigenstates $\psi_{n,m}$, and $E_n - E_m$ is the minimum energy gap [24]. It can be seen that a smaller energy gap requires a slower evolution in order to remain adiabatic, and that for degenerate energies ($\Delta E = 0$) no adiabatic evolution is possible.

This technique is useful as it allows particular eigenstates (usually the ground state) of a very complicated Hamiltonian (H_p) to be found simply by evolving from a Hamiltonian whose eigenstate is easy to find and prepare (H_0) [25]. Importantly, this process is universal and deterministic—if the system starts in the ground state of H_0, then it is guaranteed to be in the ground state of H_p after adiabatic evolution.

However, since a truly adiabatic process takes infinitely many steps and therefore an infinite amount of time, this is not possible in practice. Instead, the adiabaticity condition of AQC can be relaxed to allow a shortening of the evolution time—this is quantum annealing[2]. Over these shortened timescales ($T \sim$ µs), the process is now heuristic and the eigenstate after evolution is no longer guaranteed. The criterion given by Equation (2) is not met and there is a possibility that the system is excited to a higher-energy state. The advantage of this method is that a particular evolution can be run many times, sampling the distribution of final states until an acceptable outcome is found.

Quantum annealing is used to solve combinatorial optimisation problems, which are problems that require the minimisation of a function over a discrete set of variables. If H_p is defined such that its ground state encodes the most optimal solution to such a problem, then as long as the system is prepared in the ground state of H_0 (which is relatively easy), the system has a probability of being in this solution state at the end of the annealing process. In the field of computational complexity, these problems belong to a class of complex problems called NP (nondeterministic polynomial-time). A full discussion of computational complexity is beyond the scope of this study, but in brief NP problems are difficult to solve via classical algorithms as the time to solution scales exponentially with problem size (hence NP rather than P) [CITE]. Problems like these have large solution spaces with many local minima, which classically cannot be solved quickly. A common example is the travelling salesman problem: given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? As the number of cities increases, the number of possible routes the salesman could take grows exponentially, alongside the time taken for a classical algorithm to consider the new options.

[2] In metallurgical terms, annealing is the process of heating and cooling a material to alter its physical properties. Much like its metallurgical counterpart, quantum annealing allows a system to settle into a more useful final state.

In order to encode a problem, problem Hamiltonians (H_p) take the form of an Ising spin glass, a random arrangement of magnetic dipole moments (discrete variables) that can be in one of two states, typically spin-up ($+1$) or spin-down (-1) [26]. A spin glass with a vector s of N spins takes the form

$$H(s) = \sum_{i<j}^{N} J_{ij} s_i s_j + \sum_{i=1}^{N} h_i s_i \tag{3}$$

where J_{ij} are the coupling strengths between spins, and h_i are the field strengths of individual spins. The quantum equivalent can be expressed as

$$H_p = H(\sigma^z) \tag{4}$$

where we have replaced the spins with Pauli matrices. This is known as the Ising model, with the discrete variables now *qubits*—binary variables like their classical counterparts, but existing in a superposition until measurement. Initial Hamiltonians then take the form

$$H_0 = h_0 \sum_{i=1}^{N} \sigma_i^x \tag{5}$$

such that its ground state is an equal superposition of all possible states in the eigenbasis of H_p [27].

Another way of expressing problems is via the QUBO (quadratic unconstrained binary optimisation) model. A QUBO model takes the form of a function

$$f(x) = \sum_{i<j}^{N} Q_{i,j} x_i x_j + \sum_{i=1}^{N} Q_{i,i} x_i \tag{6}$$

to be minimised, where $x_i \in \{0, 1\}$ is a new vector of binary variables, and Q is an $N \times N$ upper-diagonal matrix of real weights. The off-diagonal $Q_{i,j}$ terms are known as quadratic coefficients, and diagonal $Q_{i,i}$ terms as linear coefficients. These two models are mathematically equivalent, and each can be transformed to the other via a simple change of variable

$$s_i = 2x_i - 1\,. \tag{7}$$

Whilst there are merits for each model, this study exclusively uses QUBO models to express optimisation problems.

2.1 Quantum hardware

Once a problem has been expressed with an Ising or QUBO model, it is sent to a quantum processing unit (QPU) to be solved. These units takes the form of a physical Ising spin glass: a

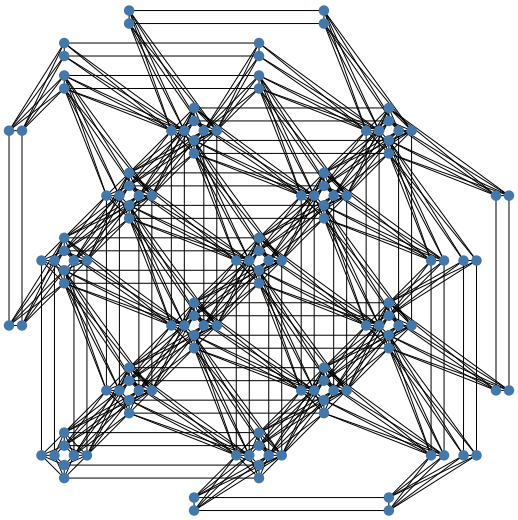


Figure 1: Graph of 144 qubits in a D-Wave QPU, using the Pegasus topology. Qubits are represented by vertices, and couplers by edges. Own work.

lattice of dipoles connected via couplers. The qubit biases and coupling values are influenced by electromagnetic fields, allowing the mapping of a model's *logical* qubits to a QPU's *physical* qubits, with linear terms as dipoles and quadratic terms as couplers between them. This mapping process is known as minor embedding, and is often handled by a classical computer "on the fly" each time a problem is submitted [28], introducing a slight computational overhead to running a problem. Fixed embeddings can be specified, but these require a priori knowledge of the specific QPU architecture. Once embedded, the system can be prepared in its initial ground state, and allowed to evolve to its final state to obtain a solution that aims to minimise the problem model.

Often, the topology of a QPU does not allow an exact one-to-one mapping of a problem to physical qubits. It may be the case that the problem requires a variable to have a higher degree (number of connections) than the number of couplers physically permitted by the QPU. The solution to this is the introduction of chains—connected physical qubits chained together, acting like a single discrete variable, which enables the necessary mapping [28]. A parameter called chain strength determines how strongly the chain is coupled, enforcing all qubits within the chain to have the same value in order for it to remain discrete. Chain strength is an important parameter that can affect the quality of solutions, and can be tuned. If it is too weak, a final state may include chain breaks: chains where the interconnected qubits differ in value. Conflicts like these can be solved via different approaches, a common one being majority vote, which sets the value of the corresponding discrete variable to the modal value of the chain. However, since this operation is done by a classical computer after the annealing process, the determined value may result in an undesirable, sub-optimal solution, and so chain breaks are often best avoided. Alternatively, if the chain strength is too strong, it can overpower the other terms in the problem model, resulting in poor optimisation [29].

QPU annealers (also known as solvers) have many other parameters that can be tuned, influencing exactly how the annealing process is carried out and significantly affecting the quality of solutions [30]. Aside from chain strength, others that this study considers are anneal time per sample and number of reads. Both these parameters affect the total problem solve time, and so finding a balance between them is key to minimising the time to solution.

As mentioned previously, the time interval for each sample (evolution) is short in order to relax the adiabatic condition. For simple problems this is sufficient to return optimal solutions with a high likelihood, however, as the problem size increases this may no longer be the case. As the number of variables increases, the energy gaps between the ground and first excited state get exponentially smaller [31]; as seen in Equation (2), smaller spectral gaps (ΔE) require longer anneal times (T) to prevent excitation to higher energy states. A benefit of shortened anneal times is that the system can be "read" (i.e. evolved and measured) many times. A problem with N qubits has 2^N energy levels that form a distribution, which is sampled each time the system is read. Increasing the number of reads raises the probability that a low-energy state from the distribution is measured.

Many companies provide both personal and commercial access to gate-based quantum computers [11, 12], however, few develop the hardware necessary for quantum annealing. D-Wave Quantum Systems Inc. ("D-Wave") was the first to realise a true quantum annealer [32], and has since developed and released their technology to the wider community [33]. All problems in this study were solved by D-Wave's Advantage System 4.1 processor[3], which boasts a topology of 5640 qubits each with a maximum degree of 15 [34]. An example of this topology can be seen in Figure 1. These computers are held in low-magnetic field environments at cryogenic temperatures to prevent unwanted fluctuations that would affect fidelity. Interaction with D-Wave QPUs is handled through their Leap quantum cloud service [35], which is used to submit problems to solvers. D-Wave provides an open-source software development package (SDK) of Python tools which allows users to create optimisation problem models for the QPU to solve.

Quantum annealing has already been applied to a widge array of optimisation problems: this study explores a novel creative application of this computing technique—music arrangement.

3 Music arrangement

Music has become a hallmark of the human experience—so much so that the *Voyager 1* probe carries 90 minutes of music from across the world as a message to potential extraterrestrial life [36]. Whilst some musicians compose entirely new music, others adapt previously-written pieces for practical or artistic reasons, whether that be in terms of instrumentation, medium, or style. This is *arrangement*. Traditionally, the arrangement of music is a complex process

[3] Access kindly granted by the Durham University Physics Department.

that requires a deep understanding of musical theory. Arrangers use their skill and creativity to create a piece that is both musically interesting and emotionally engaging, whilst still remaining faithful to the source material—a challenging and often time-consuming process. Perhaps it is unsurprising, then, that there has been interest in automating this process.

One of the earliest examples of the attempted automation of arrangement can be seen in the *Musikalisches Würfelspiel* ("musical dice game") system popular in the 18th century [37]. The roll of dice would determine the order of pre-composed musical phrases, arranging them into new combinations without the need for a composer. This system was engaged by the likes of Bach and Mozart, although fell out of fashion the following century. The introduction of computers in the 20th century allowed for more sophisticated methods of music arrangement. Composers could now transpose and manipulate musical parts digitally, without the need to transcribe parts by hand. Moving into the 21st century, more advanced techniques such as genetic algorithms and neural networks have been used to arrange music, with varying degrees of success [37]. The rearrangement of a piece can have a great effect on how the listener interprets it, and doing this effectively is still a pertinent issue many musicians face.

In this study, we aim to formulate music arrangement as an optimisation problem, which can be solved via quantum annealing. It has been shown that music arrangement can be reduced to an NP problem [38], by considering the special case of music *reduction*. Reduction is a form of arrangement whereby the number of instrument parts only gets smaller, and is defined here subject to the following conditions:

Condition 1 (Uncreative)**.** All music must come from the original, in the same order.

Condition 2 (Non-degenerate)**.** Music played in one part cannot be played in another.

Condition 3 (Monophonic)**.** Each part can only play music from one original part at a time.

Holistically, reduction aims to fit as much original music into the arrangement as possible, whilst still being playable. Notably, these constraints forgo the need to compose new music, a creative endeavour that is very subjective. This study will focus wholly on music reduction defined by these constraints.[4]

The beginning of any arrangement process is an original musical score. There exist a myriad of musical styles, genres, instruments, and notations, each as expressive and meaningful as the next. To maintain a manageable scope, this study focuses on Western classical small ensemble and orchestral music, written in modern staff notation. A brief taxonomy of a typical score is as follows: a standard score is split into parts, which can be played by a single instrument or an instrument section. Each part is further divided into bars (sometimes known as measures) which contain the notes that the instrument plays. If an instrument plays more than one note at a time, it is known as polyphonic (otherwise monophonic)—for the purposes of this study, all

[4]This is not a universal definition of reduction, but one that is useful for this study.

instruments will be treated as monophonic.

Whilst the automation of music might seem irreverent to some, the application of each new technology, from the printing press to the computer, heralds a new way to preserve the continuation of tradition. The next section outlines an original framework by which to arrange music quantumly.

4 Proposed framework

Previous approaches to the automatic arrangement of music have relied on classical methods in order to produce arrangements that are musically sound. However, these methods are limited by the complexity of the problem and the need for extensive training data. In this study, we propose a new framework for the arrangement of music via quantum annealing, an application which has not been studied before at time writing. A toy example of the proposed method can be seen in Figure 2.

4.1 Score preprocessing

Before reduction can begin, a musical score must undergo preprocessing to ensure the produced arrangement will fulfil the conditions outlined in Section 3. In particular, Condition 3 demands that all music must be strictly monophonic. The score is broken down into its constituent parts, bars, and notes—each note is represented by a vector of features, such as pitch and duration, as well as its position within the wider piece, which can be measured as an offset (the number of beats from the start of the score). Some instruments (e.g. strings) are capable of playing multiple notes at a time (chords), whereas others (e.g. woodwind) physically cannot. Additionally, composers will often write several voice parts for orchestral sections, which can be played by individuals within the section simultaneously. Both the cases of chords and voices break this condition, and so without knowing how phrases will be assigned, all music must therefore be reduced to monophony to ensure universal playability. We remove all but a single note (in this case the one highest in pitch) from chords and multi-part voicings within the score before moving onto the next stage.

4.2 Phrase identification

First, the music needs to be quantised. This could very easily be done by using predefined elements such as bars or notes, but these units on their own lack any intrinsic musical meaning. Similar to how a sentence might be split up into lexical phrases, instead of individial words or syllables, a line of music can be segmented into musical phrases, each representing a contained melodic or rhythmic idea. By preserving these smallest units of melodic lines and harmonies, we maintain the essence and familiarity of the original score, which is one of the core principles

of arrangement.

The approach taken in this study is the local boundary detection model (LBDM) [39], which aims to identify the boundaries between phrases by calculating the degree of change between successive notes; larger differences between notes would show an increased likelihood of a boundary, exploiting the fact that the starts and ends of phrases are usually characterised by a high degree of variation [39]. The strength of a boundary on a particular note is calculated over two of its parameters: pitch, and inter-onset interval (IOI), which is the time until the next note.[5] The boundary strength $S_{x,i}$ for a particular parameter x at note i is given by

$$S_{x,i} = x_i(r_{i-1,i} + r_{i,i+1}) \tag{8}$$

where $r_{i,i+1}$ is the degree of change of a parameter between notes i and $i+1$, given by

$$r_{i,i+1} = \frac{|x_i - x_{i+1}|}{x_i + x_{i+1}}\,. \tag{9}$$

In this way, a note with a high boundary strength would be very different to adjacent notes and signal the start of a new musical phrase. The set of strengths S_i for each parameter is normalised to $[0, 1]$ via

$$S'_{x,i} = \frac{S_{x,i} - \min(S_i)}{\max(S_i) - \min(S_i)} \tag{10}$$

to ensure the analysis remains generalised across all pieces. Finally, to find the total boundary strength, the strengths of each parameter are summed with a weighting, using weights derived by trial-and-error ($1/3$ for pitch and $2/3$ for IOI). The balance between these two parameters is a matter of taste and can be changed based on the perceived accuracy of the identified boundaries. If a note's total boundary strength surpasses a specified threshold, it is considered a boundary and marks the start of a new phrase. Boundaries are always taken at the beginning and end of a score.

This model is run on each instrument part in a score; once a list of boundaries is created, the phrases can be defined by taking all notes between successive boundaries. Each phrase is labelled according to the part it belongs to and its phrase index within that part, allowing the phrases to be easily referenced and reconstructed into a new score once the optimisation is complete. An example of the output of the LBDM can be seen in Figure 2a.

4.3 Problem formulation

How can the arrangement of these phrases, in fulfilment of the constraints outlined previously, be expressed as an optimisation problem that can be solved via quantum annealing? There are

[5] "Music is the space between the notes as much as the notes themselves; it's defined as much by silence as by sound." (Claude Debussy)

many pre-existing NP problems that can be solved in this way, including Boolean satisfiability and set packing [26]. Here we show that the music arrangement problem can be reduced to a graph theory problem known as *proper vertex colouring*.

A graph G can be defined as $G = (V, E)$, where V is a set of vertices (or nodes) and E is a set of edges that defines relationships between the vertices. Vertices are considered "adjacent" or "connected" if they have an edge between them. These constructions are useful to model pairwise relations between objects, and there exist a number of graph optimisation problems, each with a variety of applications.[6] Proper vertex colouring is a problem in the category of graph colouring.

Definition (Proper vertex colouring)**.** The assignment of colours to the vertices of a graph such that no two adjacent vertices share the same colour.

In this case, each phrase is represented by a vertex, with edges connecting phrases that overlap (i.e. play at the same time). The assignment of "colours" to these vertices then represents the part in the arrangement that plays the phrase. An example of a graph constructed in this way can be seen in Figure 2b. This can be seen to fulfil the necessary conditions: Condition 1 is met by only considering phrases identified from the original score; Condition 2 is met by only allowing each vertex at most one colour, preventing a phrase being played twice; Condition 3 is met by the adjacent colours constraint, meaning a part cannot play two overlapping phrases simultaneously (and therefore becoming polyphonic).

If we define $x_{v,i}$ to be a binary variable such that

$$x_{v,i} = \begin{cases} 1 & \text{if vertex } v \text{ is colour } i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

for a set of n colours, then the QUBO model for proper vertex colouring can be expressed as

$$f(x) = A \sum_{v \in V} \left(s_v - \sum_{i=1}^{n} x_{v,i} \right)^2 + B \sum_{(u,v) \in E} \sum_{i=1}^{n} x_{u,i} x_{v,i} \tag{12}$$

where A, B are the Lagrange parameters, and we have introduced the slack variables $s_v \in \{0, 1\}$.[7] The first term implements the single-colour constraint (or vertex constraint) by increasing the energy by A each time a vertex is coloured more than once ($\sum_{i=1}^{n} x_{v,i} > 1$). The nature of a reduction implies that not all phrases will be played, so the inclusion of the slack variables (specifically when $s_v = 0$) allows a vertex not to be coloured at all. Similarly, the second term implements the colour-adjacency constraint (or edge constraint) by increasing the energy by

[6] The travelling salesman problem is often represented as a graph theory problem, with vertices representing cities and edges the routes between them.

[7] This reduces to the maximum independent set problem when $n = 1$.

B for each pair of identically-coloured vertices connected by an edge (i.e. $\sum_{i=1}^{n} x_{u,i}x_{v,i} = 1$). When this function is minimised (strictly zero, in this case), the proper vertex colouring problem is solved and all the conditions are met for a valid arrangement. The $x_{v,i}$ can then be read off from the solution and cross-referenced with their corresponding phrases to give the final score, an example of which can be seen in Figure 2c.

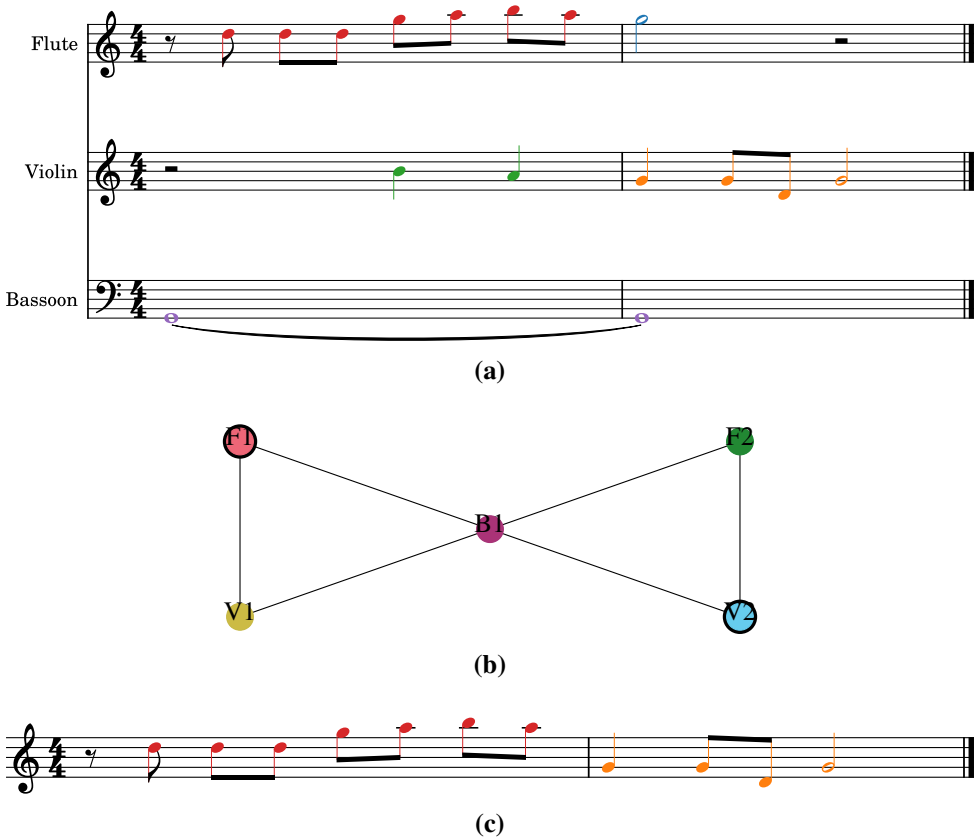


Figure 2: A toy example demonstrating a simplified version of the proposed framework. **(a)** An example score of three parts. Phrases identified by the LBDM have been outlined. **(b)** A graph of the score, with the colour and label of each vertex corresponding to its phrase, and edges connecting overlapping phrases. One possible solution of proper vertex colouring, with $n = 1$, is shown by the outlined vertices. **(c)** The score representation of the final arrangement.

4.4 Musical entropy

Despite having a valid arrangement of phrases, this does not guarantee that the music produced will have any sense of "musicality". To ensure the arrangement is as musically interesting as possible, each vertex of the graph can be weighted to introduce a bias according to its phrase's "musicality". The idea of a musical "utility" has been used before with classical reduction algorithms [40] and to assess playability rules [41]. Here we quantify this by calculating a phrase's *musical entropy*.

Entropy can be seen as how much information a random variable contains; in this context, a phrase with a higher entropy would indicate a higher level of variation and complexity, which

can be seen as a likelihood of it being more "musical" [42]. Maximisation of phrase entropy would therefore result in the creation of richer arrangements. For a discrete random variable X with a probability distribution $P(X)$, the Shannon entropy is defined as

$$H(X) := -\sum_i P(x_i) \log_2 P(x_i) \tag{13}$$

where each x_i is a possible value of X and $\sum_i P(x_i) = 1$. In this context, the random variable is a musical note, considering its possible values in terms of both pitch and duration.

Pitch entropy is calculated by considering the distribution of pitches in a phrase. The probabilities of each pitch x_i can be found by

$$P(x_i) = \frac{n_i}{N} \tag{14}$$

where n_i is the number of times pitch x_i appears in the phrase and N is the total number of notes in the phrase. A phrase with a greater variety of pitches (and thus greater entropy) will likely be more "interesting" and so should have a higher bias to be included in the final arrangement. Rhythm entropy is calculated in a similar manner, but instead considering x_i as possible duration values. The total entropy of a phrase is then the sum of the pitch and rhythm entropies combined.

We can modify the QUBO model of Equation (6) to consider vertex weights by adding an objective term that looks like

$$f_C(x) = -C \sum_{v \in V} \sum_{i=1}^{n} W_v x_{v,i} \tag{15}$$

where W is a vector holding the weights for each vertex, and C is the associated Lagrange parameter. Objective terms are negative to ensure than the minimisation of the function results in the maximisation of the objective (here the sum of the total weights).

Weights can also be added to edges, to bias the selection of pairs of vertices. Here we assign edge weights as the absolute difference between the weights of the vertices it connects, adding a tendency to colour connected vertices that have very different weights. Musically, this means simultaneous phrases are more likely to be picked if their musical entropies differ; complex, strongly varying, high-entropy phrases may sound noisy and confusing if played together, so this choice allows each high-entropy phrase to be accompanied by a simpler, low-entropy one. The associated objective term modification to Equation (6) looks like

$$f_D(x) = -D \sum_{(u,v) \in E} W_{uv} \sum_{i=1}^{n} \sum_{i<j}^{n} x_{u,i} x_{v,j} \tag{16}$$

where W is now a matrix holding the weights for each edge, and D is the Lagrange parameter.

Finally, we introduce a bias towards the assignment of vertices to specific colours, based on the instrument the colour represents. It has been shown that instrument parts within a score can be categorised into distinct "arrangement elements", or roles, which reflect their musical function within the piece [43]. Here we consider only three: lead, often including melodic or countermelodic lines; rhythm, providing harmony; and foundation, the fundamental bass line. Rudimentarily, we can classify phrases into these three categories by their entropy. Melodic lines often vary more, have higher entropy, and therefore should be classed as lead, whereas lower-entropy phrases should tend towards rhythm or foundation. By defining threshold entropy values for each category, phrases can be given a bias towards instruments that have been assigned that category. For example, a flute part assigned lead in the arrangement should contain more high-entropy phrases than a cello part assigned as foundation. The corresponding objective term takes the form

$$f_E(x) = -E \sum_{v \in V} \sum_{i=1}^{n} \theta(W_v - W_{\text{th},i}) x_{v,i} \tag{17}$$

where $W_{\text{th},i}$ is the threshold value for the category of instrument i, θ is the Heaviside step function, and E is the Lagrange parameter. In this study we take $W_{\text{th},l} = 2\max(W)/3$ to be the threshold value for lead and $W_{\text{th},r} = \max(W)/3$ the threshold for rhythm (where $\max(W)$ denotes the largest entry of the weight matrix), in order to divide phrases equally between the three categories (assuming a uniform distribution of entropy).

Taking these functions together, the final QUBO model is

$$\begin{aligned} f(x) &= f_{A,B} + f_C + f_D + f_E \\ &= A \sum_{v \in V} \left(s_v - \sum_{i=1}^{n} x_{v,i} \right)^2 + B \sum_{(u,v) \in E} \sum_{i=1}^{n} x_{u,i} x_{v,i} \\ &\quad - C \sum_{v \in V} \sum_{i=1}^{n} W_{v,v} x_{v,i} - D \sum_{(u,v) \in E} W_{uv} \sum_{i=1}^{n} \sum_{i<j}^{n} x_{u,i} x_{v,j} - E \sum_{v \in V} \sum_{i=1}^{n} \theta(W_v - W_{\text{th},i}) x_{v,i} \end{aligned} \tag{18}$$

which requires $N(n+1)$ variables, where N is the total number of phrases.

4.5 Lagrange parameter analysis

To ensure that the minimisation of Equation (18) does indeed fulfil the constraints necessary for a valid arrangement, the Lagrange parameters must be tuned. The very nature of the problem means a full parametric analysis would be complicated, so this study focuses on an empirical approach. Nonetheless, a quick algebraic calculation can help reduce the problem. The model

given by Equation (18) contains two constraints, each which aims to fulfil a different condition. The vertex constraint (labelled with A) upholds Condition 2, the breaking of which would introduce phrases being duplicated in the final arrangement. The edge constraint (labelled with B) upholds Condition 3, preventing phrases that overlap being played by the same part. The number of duplicates and overlaps needs to be strictly zero in order for a solution to be valid therefore these Lagranage parameters need to be large enough to facilitate this.

In a worst-case scenario, a vertex included in the solution would break both these constraints—to ensure that this results in an overall increase in energy, and is therefore less likely, we can derive the inequality

$$A + B > C \max(W) + D \max(W) + E \,. \tag{19}$$

To allow a similar analysis across problems with different weight matrices, we set $C = D = 1$ and introduce new parameters X_m that scale with the maximum weight, such that $X = X_m \max(W)$, giving us the expression

$$A_m + B_m > 2 + E_m \,. \tag{20}$$

Throughout this study we set $E_m = 1$ so that the bias towards part roles is present but subtle. The values of A_m and B_m can then be found by performing parametric variation and measuring the number of broken constraints (duplicates and overlaps) in the lowest-energy solutions. Parameter values should be just large enough to fulfil Equation (20), whilst not being too large as to overwhelm the other terms in the QUBO equation [CITE].

4.6 Solution interpretation

Once the problem has been submitted to a quantum annealer and a distribution of solutions returned, we can finally construct the arrangement. Out of all the samples, the lowest-energy valid (i.e. no duplicates or overlaps) sample is picked to be the solution. The binary variables assigning vertices to colours can be read off and interpreted as phrases and instruments, respectively, allowing the music of the original score to be transformed into the final arrangement. At this point, music post-processing can also be applied. In this study, phrases are shifted up or down octaves depending on the pitch range of the target instrument, to ensure playability. An example of an arranged score can be seen in Figure 2c.

5 Results

The outlined framework was applied to two scores: Quartet No. 1 in B-flat major, Op. 1, by Joseph Haydn (hereon referred to as Haydn Op. 1), and Symphony No. 5 in C minor, Op. 67, I. Allegro con brio, by Ludwig van Beethoven (referred to as Beethoven Op. 67). Haydn Op.

1 is of a Baroque style, consisting of 63 bars with four instrument parts that play consistently throughout the piece, structured into short, well-defined phrases. This score was chosen as it was expected that the LBDM would be reasonably successful in discretising the piece, and that its short length and smaller instrumentation would result in a manageable number of variables to be embedded into the QPU. Beethoven Op. 67 is of a Romantic style and was shortened to the first 21 bars, with 12 instrument parts that are introduced at different times in the piece. This score was chosen for limit-testing purposes, as its larger instrumentation greatly increases the connectivity of the problem and the size of the embedding. Both pieces are relatively well-known meaning their constructed arrangements could be criticised more readily by ear.

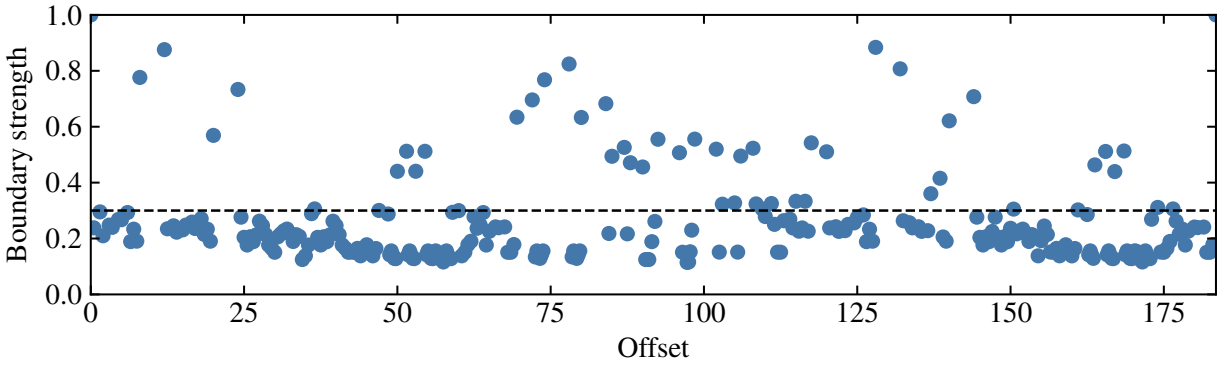


Figure 3: Calculated boundary strengths for the Violin I part of Haydn Op. 1. The threshold of 0.3 is denoted by the dashed line, resulting in 51 identified boundaries.

An example of the output of the LBDM can be seen in Figure 3, here applied to the Violin I part of Haydn Op. 1, with a threshold of 0.3. Thresholds were chosen manually by examining the distribution of boundary strengths, and keeping in mind the size limitations of embeddings. Notes with boundary strengths above the chosen threshold were taken as boundaries, defining the phrases that would become the vertices of the problem graph. Overall this resulted in 192 identified phrases for Haydn Op. 1, and 127 phrases for Beethoven Op. 67 (using a threshold of 0.25). The entropy of each phrase was calculated using Equation (13), to be used as the corresponding vertex weight and to calculate the edge weights, and the QUBO model was then constructed using Equation (18).

Once the QUBO models had been created, they were embedded into the QPU architecture. The scaling of the number of qubits required with the number of arrangement instruments for both pieces can be seen in Figure 4. It's important again to differentiate between *logical* qubits, which are the variables that define the problem, and *physical* qubits, which exist in the QPU itself onto which the problem is embedded. The number of logical qubits required is well-defined as $N(n+1)$ where N is the number of phrases and n the number of instruments, and so increases linearly with n (except for $n = 1$ in which case only N qubits are required since slack variables are not needed). In contrast, the number of physical qubits increases more quickly, shown here with a quadratic fit. We are reminded that the maximum degree of a qubit in a D-Wave QPU is 15, that is, each qubit can only connect with 15 other qubits; any more than

this and we require qubits to be chained. The difference between physical and logical qubits can be seen as the number of extra qubits required in chains to provide the desired connectivity. Beethoven Op. 67 shows this difference significantly: despite requiring fewer logical qubits than Haydn Op. 1 (owing to its smaller number of phrases), the number of physical qubits required is consistently higher. The original score contains over twice the instrument parts, meaning its phrases have more overlaps needing a greater degree of connectivity from its variables. Due to this rapid increase in physical qubits, Beethoven Op. 67 models could not be embedded onto the QPU for $n > 4$, although the projected number of qubits required has been shown on the plot. Haydn Op. 1 embeddings were comfortably below the qubit limit (shown by the dotted line) up to the piece maximum of $n = 4$. Subsequently, Haydn Op. 1 was chosen to be arranged for up to four instruments, but instead as a traditional woodwind quartet (Flute, Oboe, Clarinet, Bassoon). Beethoven Op. 67 was also chosen to be arranged for up to four instruments, but now as a string quartet (Violin I, Violin II, Viola, Violoncello). Full details can be found in Appendix B. Once these embeddings were calculated, the problems could be sent to the QPU for solving.

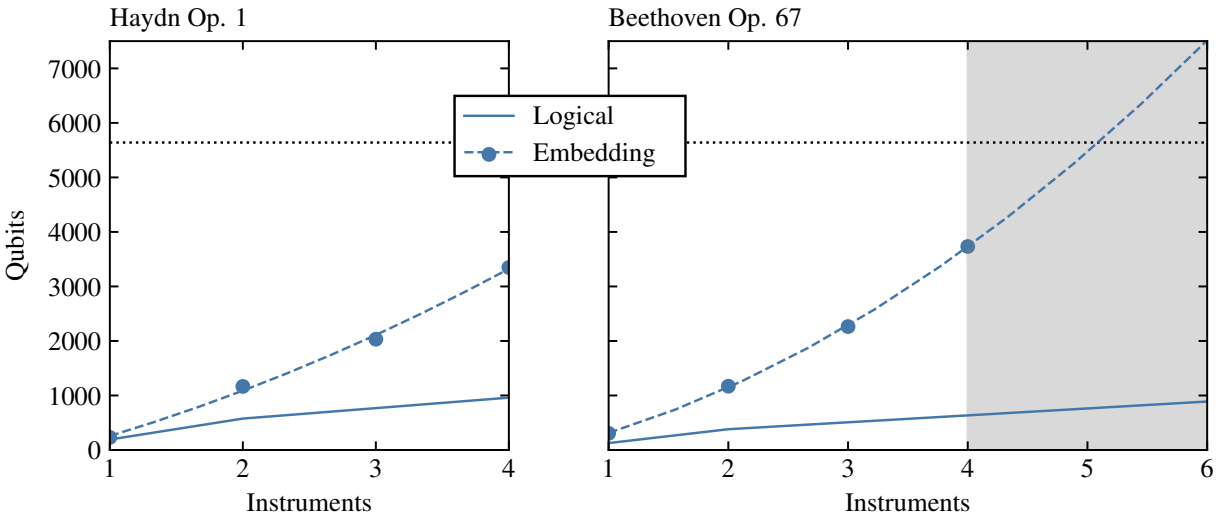


Figure 4: Number of qubits required for problems of different instruments, with the dotted line marking the maximum number of accessible qubits in a D-Wave QPU (5640). Logical qubits (equivalent to the number of variables) shown as solid lines, physical qubits required to embed the problem shown as points with a dashed line quadratic fit. Shaded region represents a theoretical extrapolation to arrangement for a higher number of instruments.

An example of the energy distribution of a returned sample set can be seen in Figure 5, using Haydn Op. 1 with $n = 3$. As the number of reads tends to infinity, this distribution should reflect the true distribution of the energy eigenvalues of the optimisation problem, and at 2000 reads already seems to be well-sampled.[8] The most optimal solutions lie in the leftmost tail of the distribution, corresponding to the lowest energies. The distribution is very symmetric, meaning the probability of returning these low-energies is not favourable, but not impossible.

[8]For this particular problem there are $2^{768} \sim 10^{231}$ levels. To put that into perspective, the Universe is said to contain about 10^{24} stars.

The spectral gap distribution can also be seen, which measures the difference in energy between adjacent eigenstates. It takes the form of an exponential decay, with many samples having very small (< 0.1) energy gaps, implying that many of the energy eigenstates are degenerate. According to Equation (2), depending on the location of these small energy gaps, this could mean that longer evolution times are required to remain in a low-energy state and find optimal solutions.

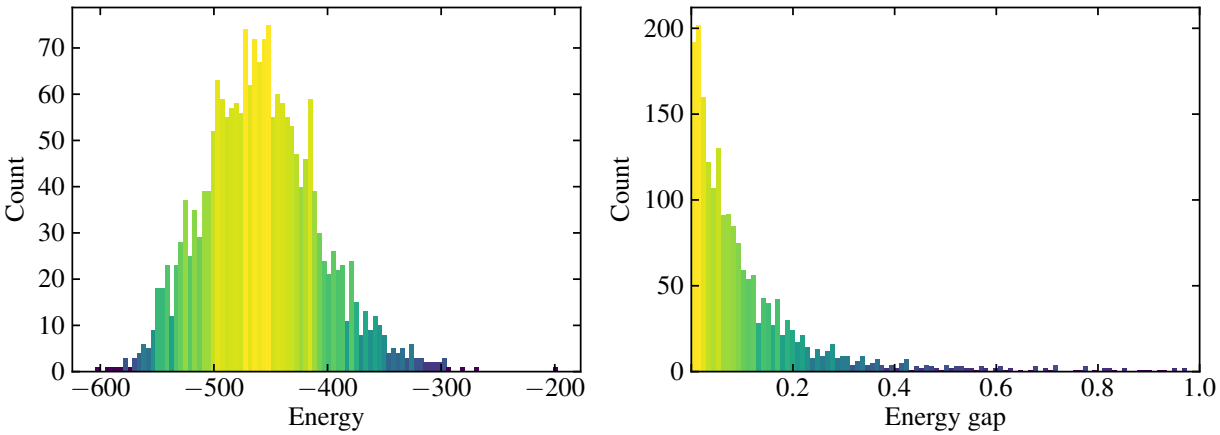
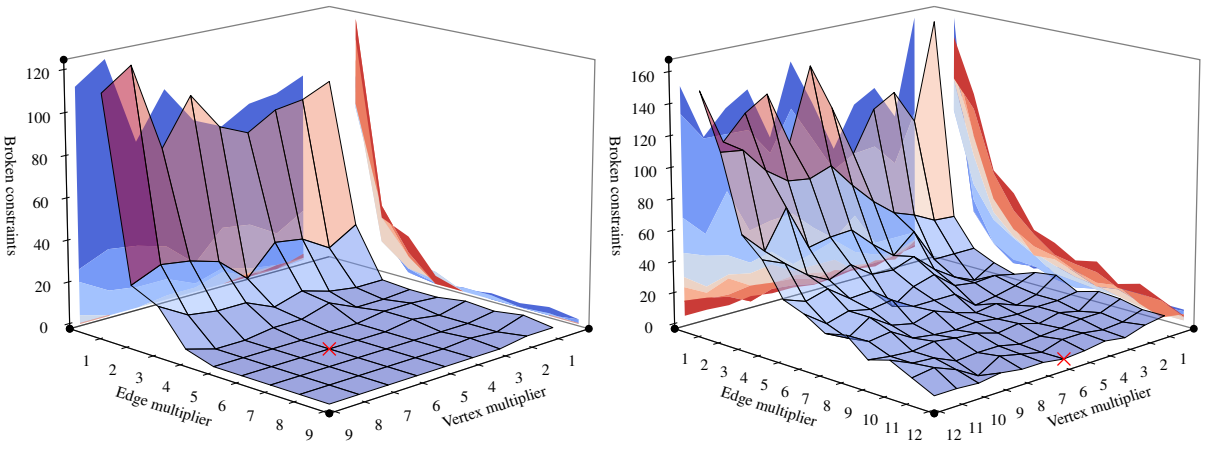


Figure 5: Example of a returned sample set from solving the Haydn Op. 1 problem for $n = 3$, 2000 reads. **(Left)** Histogram of the sample energies. **(Right)** Histogram of the gaps in the energy spectrum (gaps greater than 1.0 not shown).

5.1 Parameter tuning

Next, the Lagrange parameters A_m and B_m were tuned, the results of which can be seen in Figure 6. These two parameters were varied together as the complexity of the QUBO model with such a large number of variables makes it hard to predict how the terms interact. The number of broken constraints was taken as the sum of overlaps and duplicates combined. In terms of reducing this number, the edge multiplier (B_m) can be seen to have the most importance, although an increase in the vertex multiplier (A_m) is still required to reduce this number to exactly zero in order to produce a valid solution. The parameters used for subsequent models were $A_m = 6, B_m = 6$ for Haydn Op. 1 and $A_m = 6, B_m = 12$ for Beethoven Op. 67, which can be seen to fulfil Equation (20) as expected. The higher edge multiplier required for Beethoven Op. 67 can be seen as an effect of its increased connectivity; the original score has over twice the number of instrument parts, increasing both the number of edges in the problem graph and the number of constraints that could possibly be broken. The tuning of these parameters does not necessarily guarantee that no constraints are broken in the lowest-energy solutions, however, they will be less likely.

Optimisation of both the chain strength and anneal time per sample solver properties can be seen in Figure 7. For both scores, increasing the chain strength leads to a direct decrease in the fraction of chains broken. However, the chain break fraction does not necessarily need to be



(a) **(b)**

Figure 6: Parametric plot tuning the Lagrange parameters A_m (vertex multiplier) and B_m (edge multiplier). The QUBO equation for each pair of parameters was solved 10 times, and the mean number of broken constraints in the lowest-energy solution taken. Chosen values marked with a cross. **(a)** Haydn Op. 1 parameters varied up to a maximum of 9, with $A_m = 6, B_m = 6$ chosen. **(b)** Beethoven Op. 67 parameters varied up to a maximum of 12, with $A_m = 6, B_m = 12$ chosen.

exactly zero for valid solutions to be returned. Chain breaks increase the likelihood of broken constraints, as they are resolved after the evolution process and may not reflect a minimisation of the QUBO problem, but as long as the fraction is small then there is still a good chance the lowest energy solutions can still be valid. Both scores also demonstrate a minimum point in the lowest energies of the returned sample sets. Before this point, the chain strength is too weak to enforce the homogeneity of the chain, leading to more chain breaks and increased energy from their resolution. After this point, the chain strength overwhelms the other terms in the QUBO model, resulting in a priority to maintain chains, at the detriment of the minimisation of objective terms. In order to choose a suitable chain strength, a compromise has to be made between the minimisation of the energy and the fraction of broken chains: a chain strength of 25 was chosen for Haydn Op. 1, and 30 for Beethoven Op. 67, to meet this compromise. The embeddings for Beethoven Op. 67 have longer chains in general (see Figure 4), due to the higher connectivity of the QUBO model, so its higher chain strength accounts for the increased difficulty in keeping longer chains single-valued.

Anneal time per sample could only be increased to a maximum of 300 µs for a problem submission of 1000 reads due to time limits imposed by the solver. In theory, as $t \to \infty$, the energy of the samples should tend to some minimum ground state energy due to the adiabatic theorem; here, we do observe a general decreasing trend, but not one that is particularly steady, especially for Beethoven Op. 67. It may be that the solutions found are already close to the true minimisation of the QUBO problem (but unlikely at such short time scales), or that there is

some local minimum due to small spectral gaps that requires a longer anneal time to resolve, as in Equation (2). In the interest of time and the desire to increase the number of reads, an anneal time per sample of 200 µs was chosen for both Haydn Op. 1 and Beethoven Op. 67.

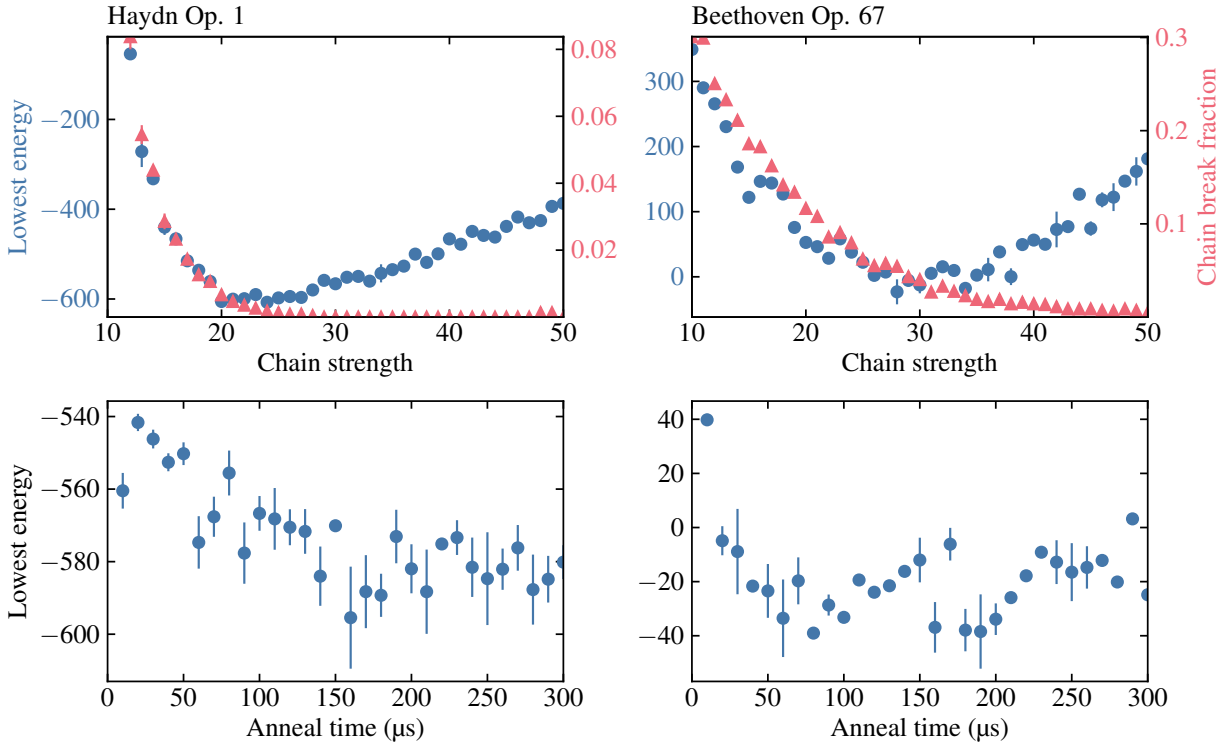


Figure 7: Optimisation of the solver configuration. Each data point was sampled five times, with the mean and standard error calculated. **(Top)** Variation of the fraction of chains broken (squares) and energy (circles) with chain strength of the lowest energy samples in a sample set of 1000 reads, using the default anneal time (20 µs). **(Bottom)** Variation of lowest energy with anneal time per sample in a sample set of 1000 reads, using the updated chain strength.

5.2 Comparison to classical algorithms

Having tuned the Lagrange parameters and solver configuration, we compare returned solutions from the QPU to classical optimisation algorithms solving the same problem. D-Wave provides a selection of heuristic classical samplers that can return solutions to QUBO problems, two of which are used in this study.

The steepest descent method, the discrete analogue of gradient descent, takes steps according to a local minimisation—the dimension along which to descend is determined by the variable flip that causes the greatest reduction in energy [44]. The initial state of the solver is randomised for each sample, so although it is guaranteed to find a local minimum with enough steps, for a complex problem with many such minima the same solution won't necessarily be returned. Despite its apparent rudimentary nature, this method is still very widely used for solving optimisation problems.

Simulated annealing, on the other hand, is more complex and works similarly to quantum an-

nealing, but instead of evolving over a time interval, the evolution is instead determined by a "temperature" parameter T [45]. At each step, the probability P of moving between states depends on the difference in energy ΔE and T. At all times, $P > 0$ even if ΔE—however, as $T \to 0$, $P \to 0$ if $\Delta > 0$ and is positive otherwise. This evolution of P means that the system can initially climb potential barriers, before settling into minima as the temperature decreases. In this implementation, temperature schedule is varied between samples, changing both the initial and final T values. Due to its similarity, this method is often used as a direct comparison between quantum and classical approaches [46].

A comparison of the lowest-energy solutions returned by the different solvers for both pieces can be seen in Figure 8, whilst varying the total number of reads. The maximum number of reads for a single problem was limited to 2000 due to the solver time constraints previously mentioned. Both pieces demonstrate very little variation in lowest-energy solutions returned by quantum annealing as the total number of reads is increased. This could be a sign that the lower end of the energy distribution was already well-sampled with a small number of reads, so increasing reads would only provide marginal gains. Both classical methods also vary very little, which is to be expected due to their semi-deterministic nature. Looking at Haydn Op. 1, quantum annealing consistently provides considerably better lower-energy solutions than steepest descent, and surpasses simulated annealing by a small margin at reads upwards of 800. On the other hand, quantum annealing performs poorly at providing low-energy solutions against both classical methods for Beethoven Op. 67. This may be due to the high connectivity of the model posed by this score. Best-fit lines found via a least-squares regression suggest that the lowest-energy solutions returned by quantum annealing would surpass simulated annealing at approximately 13 200 reads, requiring a runtime over six times the current limit to be advantageous.

A similar comparison can be seen in Figure 9, but this time comparing the other model objective, musical entropy. We note that, unlike sample energy, the model aims to maximise this value. Both pieces again display very little variation in entropy with the number of reads. Similar to the comparison in energy, quantum annealing performs better than the classical algorithms in optimising the Haydn Op. 1 problem, consistently providing higher-entropy solutions, whilst performing consistently worse for Beethoven Op. 67. This is perhaps unsurprising, as energy and entropy are intrinsically linked in the QUBO model, with a minimisation in energy likely leading to a maximisation of entropy. However, the entropy value does give more information about the perceived "quality" of the solution, such as how effectively phrases have been assigned according to instrument roles.

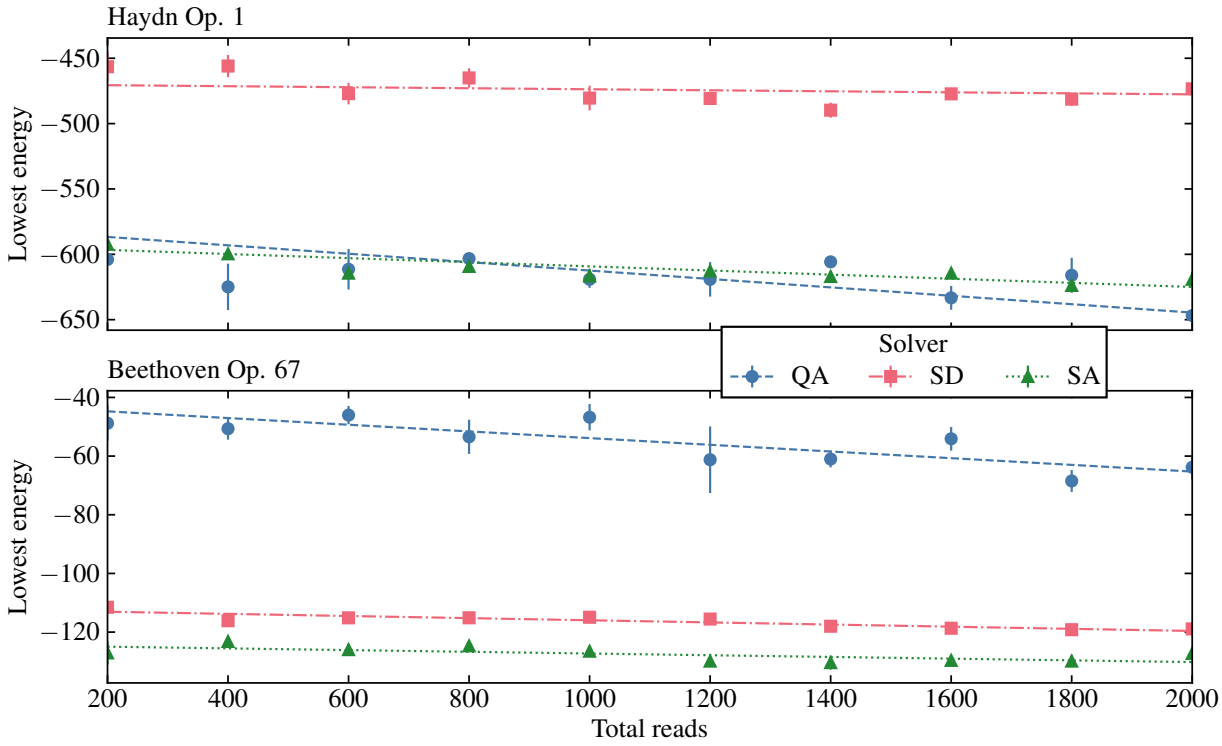


Figure 8: Comparison between quantum annealing (QA), steepest descent (SD), and simulated annealing (SA) sample set energies, varying the number of reads, at $n = 3$. Best-fit lines found via a least-squares regression.

5.3 Scaling

A comparison to classical algorithms can also be made as the problem size changes—in this case varying the number of arrangement instruments n. Increasing n not only increases the number of equation variables, but also the degree of those variables, raising the connectivity of the problem. Variation of both lowest energy and time to solution with the number of instruments can be seen in Figure 10. Focusing on the sample energies, all solvers perform similarly for both pieces when $n = 1$. This similarity ends, however, as n increases. For the problems posed by Haydn Op. 1, quantum annealing performs better than both classical algorithms until $n = 4$, at which point simulated annealing outperforms. This point can be seen as the connectivity limit at which the problems become too complicated to embed efficiently onto the QPU, with long chains resulting in an increased probability of broken chains that raise the energy. Beethoven Op. 67 models can be seen to be already at this limit when $n > 1$, with the performance of both classical algorithms only getting better compared to quantum annealing as n increases.

When measuring time to solution, only the annealing time is considered for quantum annealing (i.e. anneal time per sample multiplied by the number of reads, which remains constant), as opposed to the total QPU access time. Many other operations occur during problem submission, including programming, delays between samples, and readout, but the time this takes reflects a limitation of the technology rather than the method, so was not taken into account. Both pieces demonstrate a similar relationship, with quantum annealing being considerably faster

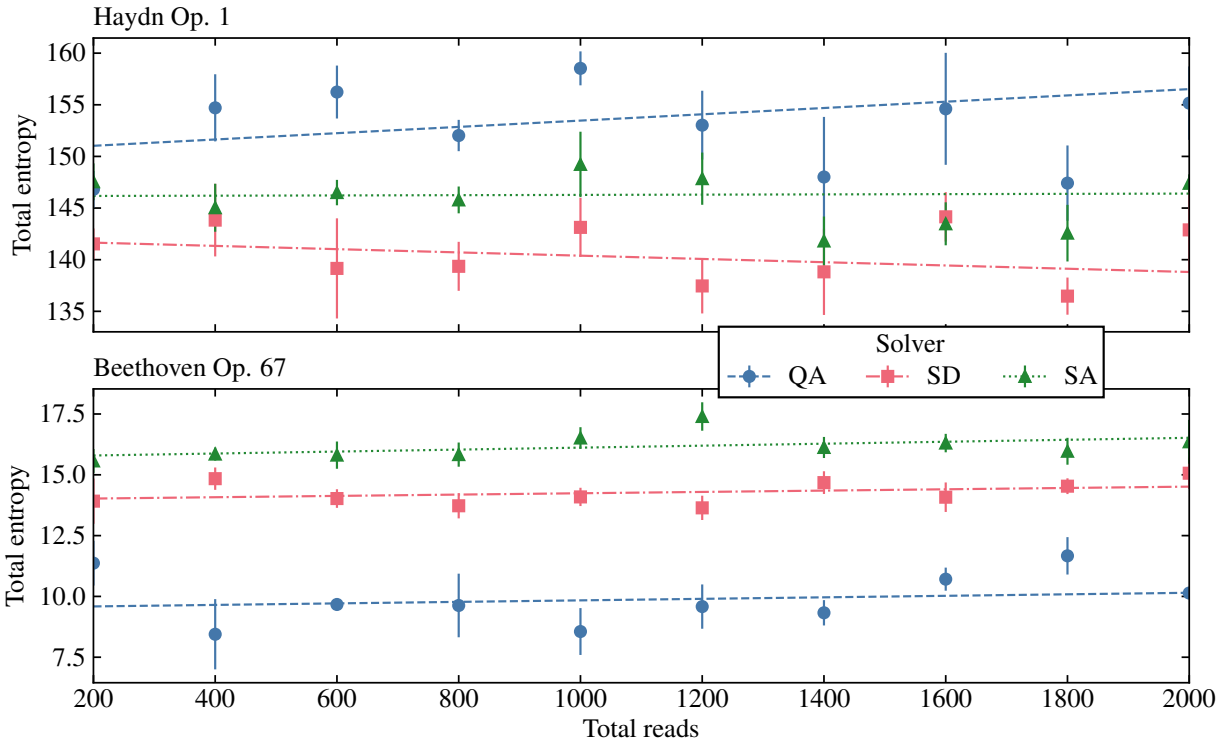


Figure 9: Comparison between quantum annealing (QA), steepest descent (SD), and simulated annealing (SA) entropies of the same lowest-energy samples as Figure 8, varying the number of reads. Best-fit lines found via the least-squares method.

than simulated annealing, but slightly slower than steepest descent. Time to solution for both classical algorithms should increase with n as the system has to consider many more states within the solution space. The simplicity and speed of steepest descent results in very little variation, but with simulated annealing this effect is much more significant. This can be seen as an advantage of quantum annealing, as its time to solution remains constant.

Throughout this testing, low-energy valid solutions were sampled and converted into musical scores, which could also be used to generate audio files. As with all of the arts, subjective judgement is difficult, but we note that the generated arrangements were not unpleasant to the ear. An overview of the code used throughout this section can be found in Appendix A, and original scores, with their arrangement counterparts, can be found in Appendix B.

6 Conclusions

Holistically, it can be said that the formulation and application of the propose framework is a success. Music arrangement is readily identified and constructed as an original problem to be solved via quantum annealing, which is then run on real quantum annealers to return solutions that provide valid arrangements. For both pieces trialled in this study, the QPU was able to give valid solutions for a range of arrangement instruments (albeit some more successful than others), and, in the case of Haydn Op. 1, outperform two common classical algorithms in terms

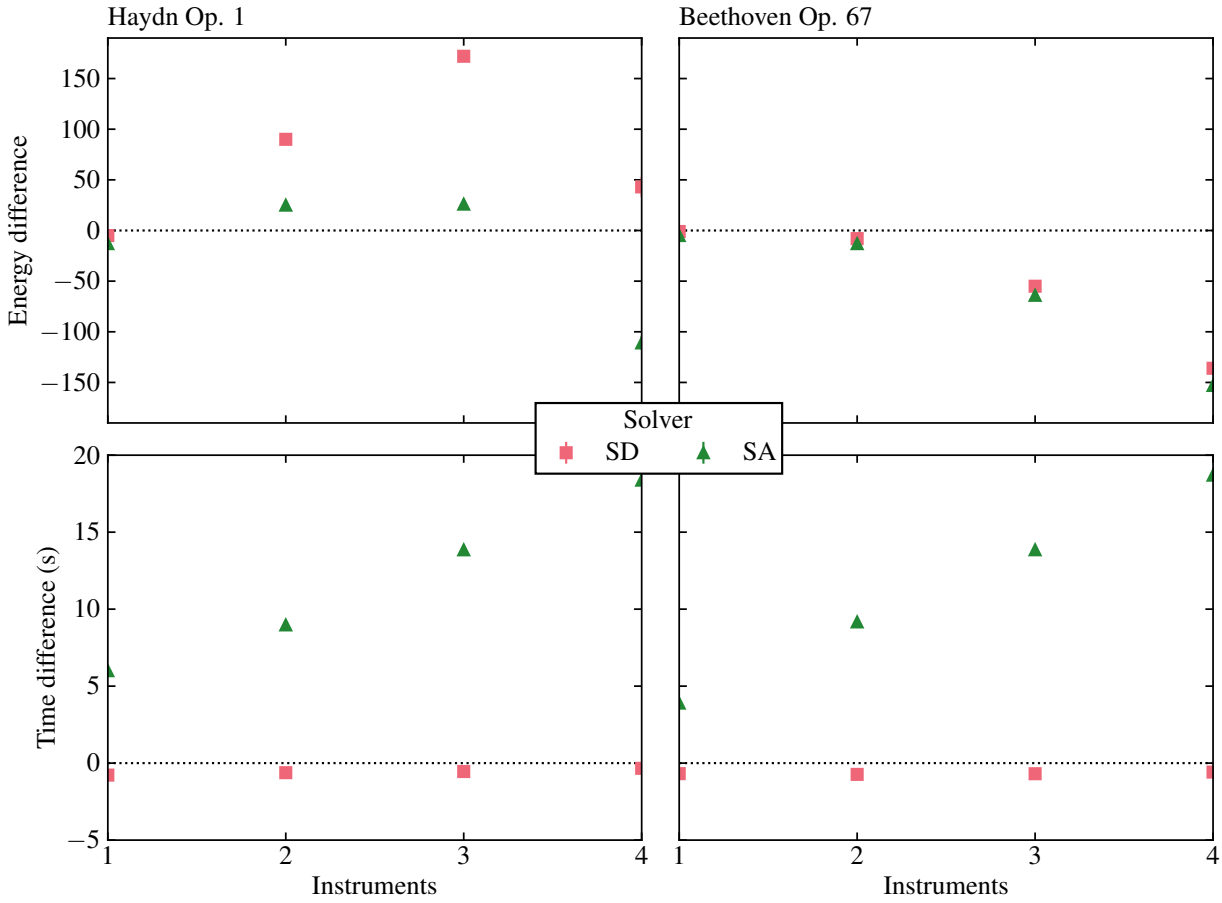


Figure 10: Comparing the solutions returned by classical solvers against quantum annealing, at 2000 reads. Difference from quantum annealing shown, with a positive difference signifying an advantage towards the quantum solver. **Top:** Lowest-energy samples of the returned sample set. **Bottom:** Time to solution, in seconds.

of solution energy and time to solution.

The nature of classical algorithms means that, for an NP problem, the time to solution increases exponentially with problem size. Given that the quantum time to solution remains constant, quantum annealing will eventually show an advantage as the problem size increases. Here we have shown this advantage in some scenarios, but even so one that is relatively small. The energy advantage seen in Figure 8 against simulated annealing is only a small fraction of the overall energy, and the time advantage seen in Figure 10 is only of the order of seconds, nowhere near the "one million years" of D-Wave's most recent claim. Bearing in mind also that the classical solvers were used "out of the box" and not tuned in any way as with the quantum method—it may be the case that proper configuration would reverse any quantum advantage gained. Additionally, there exist many other classical optimisation algorithms not considered, some which may be more suited to solving this particular problem. This study does not claim quantum supremacy—this does not mean that the application of quantum annealing has not been useful.

Nonetheless, this process has its limitations, the primary being the sophistication of current

quantum technology. Although the technology has been improving for several years, at time of writing quantum annealers are still at an early stage. D-Wave's latest Advantage 4.1 chip is their latest iteration, but at just under 6000 qubits with a maximum degree of 15, still leaves much to be desired for solving large, highly-connected problems. As seen in Figure 4, whilst Haydn Op. 1 could easily be embedded, Beethoven Op. 67 quickly overwhelmed the QPU, despite being shortened to just 21 bars, a mere 4 % of the entire piece. With current technology, both the length of scores and number of arrangement instruments are limited. D-Wave is currently experimenting with a new Zephyr topology which would allow embedding of problems with over 7000 physical qubits with a maximum degree of 20, resulting in a 2–10 % improvement in chain lengths [47]; in circulation, these QPUs would allow more complex problems to be solved. Furthermore, D-Wave imposes a time limit on problems submitted to its QPUs (at least for commercial use), alongside an overall monthly access allowance, constraining the total annealing time possible. Once quantum annealing technology becomes more commonplace, QPU time will be less scarce, allowing the possibility of longer anneal times which can resolve smaller spectral gaps.

One way to work with the technological limitations is to reduce the complexity of the problem. Graph partitioning, splitting a problem graph into subgraphs, allows a large problem to be decomposed into those small enough to be embedded, which can submitted and solved individually. The key to such a method would be to partition the graph such that the connectivity between subgraphs is minimised, to reduce potential disagreements at connecting points. There are several techniques that employ this that were beyond the scope of this study [48], but could be considered in future work. Another method is to reduce the number of ancilla (additional) qubits required to define the problem. Currently, the proposed QUBO model requires $N(n+1)$ logical variables, with that additional one being slack variables needed to define an inequality constraint (the number of colours a vertex can have is ≤ 1). An alternative to this method of slack penalisation is unbalanced penalisation, which is able to define inequality constraints without the need for extra variables by introducing new Lagrange parameters [49]. This method was not considered in this study due to the extra analytical overhead of calculating the values of these parameters, but could be incorporated into future models to reduce the overall problem size. In this same vein, the tuning of Lagrange parameters can be treated as a meta optimisation problem, with the goal of minimising the gap between the optimal solution of the original problem and the ground state of the Hamiltonian [49], although this was considered unnecessary for this study.

There exist advanced annealing techniques that could be used to further improve solutions. Here we have considered the overall anneal time per sample, but it is possible to customise the anneal schedule as well, changing the rate at which the Hamiltonian evolves [50]. This can also include short pauses midway through evolution, allowing the system to reach thermal equilibrium and subsequently return more optimal solutions [51]. Another useful technique is

reverse annealing, during which a finished evolution is annealed backwards and then forwards to a new state, allowing the refinement of a solution [52].

This framework was tested on but a small fraction of the vast collection of music available. The application of this method on a wide range of pieces from different musical styles would be instrumental in testing its versatility. Beyond the quantum aspect, the surrounding framework itself also has room for improvement. Firstly, the preprocessing of scores, removing notes from chords and additional voices in an arbitrary manner, results in the loss of information that could be useful for the final arrangement. A more advanced technique could split polyphonic phrases into multiple monophonic ones, each with its own assigned variable, increasing the size of the problem slightly for the benefit of higher musical fidelity.

The effect of LBDM parameters (pitch/IOI weighting, boundary strength threshold) on phrase identification could also be studied. Ideally, musical phrases should be fairly short and similar in length, to give the proper vertex colouring algorithm the best chance in selecting the most interesting sections of the score. This would prevent important notes being hidden in long phrases with low entropy. Further harmonic analysis could also be employed to distinguish phrases via chord progressions.

The QUBO model used throughout, as outlined in Equation (18), has a lot of flexibility for customisation. Here, we define one measure of "musicality", musical entropy, and use it to define objective functions that we want to maximise. Equally, new quantitative measures like "playability" (how easy a phrase is to physically play) could also be defined [53], and incorporated into the QUBO model to control the complexity of music produced. Refinement of the model is vital for the ultimate "musicality" of the arrangement. With regards to solution quality, there is an important distinction between how well the QPU solves the problem, and how well the QUBO model describes the problem. Any produced arrangement will only be as good musically as the model that describes it.

Although it can be argued that music cannot be objectively "scored", nonetheless, the quality of the produced arrangements should be judged in some way. One suggestion is that the "goodness" of music can only be measured via a Turing-like test, where human subjects are presented with a selection of both human- and computer-generated scores, and asked to categorise them [54]. To this extent, a good measure of the arrangements produced by this method would be to compare them against popular arrangements of the same score composed by human professionals, and by classical algorithms, via a series of blind trials. If the participants fail to identify the quantum compositions from the human or classical ones more often than random chance, then it can be said that the generated arrangements are of sufficiently good quality. We note that, whilst being pleasant, the generated arrangements would not stand up to this scrutiny in its current state.

To conclude, this study has shown the application quantum annealing to the automatic arrange-

ment of music. By formulating a musical score into a function to be minimised, the annealing process can identify parts of the original to become the final arrangement based on defined objectives and constraints. An application of this sort, whilst still comparable to classical processes, is useful to consider in the emerging and unpredictable world of quantum computing. Just as science and business do already, it can be hoped that the arts take advantage of promising new technologies as well.

References

[1] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982. ISSN: 1572-9575. DOI: 10.1007/BF02650179.

[2] J. Preskill. Quantum computing and the entanglement frontier. 2012. DOI: 10.48550/arXiv.1203.5813. arXiv: 1203.5813 [quant-ph].

[3] F. Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5.

[4] E. Pednault et al. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits. 2019. DOI: 10.48550/arXiv.1910.09534. arXiv: 1910.09534 [quant-ph].

[5] K. McCormick. Race Not Over Between Classical and Quantum Computers. *Physics*, 15:19, 2022. ISSN: 1943-2879. DOI: 10.1103/Physics.15.19.

[6] A. D. King et al. Beyond-classical computation in quantum simulation. *Science*, 388(6743):199–204, 2025. DOI: 10.1126/science.ado6285.

[7] J. Tindall et al. Dynamics of disordered quantum systems with two- and three-dimensional tensor networks. 2025. DOI: 10.48550/arXiv.2503.05693. arXiv: 2503.05693 [quant-ph].

[8] L. Mauron and G. Carleo. Challenging the Quantum Advantage Frontier with Large-Scale Classical Simulations of Annealing Dynamics. 2025. DOI: 10.48550/arXiv.2503.08247. arXiv: 2503.08247 [quant-ph].

[9] W. G. Unruh. Maintaining coherence in quantum computers. *Physical Review A*, 51(2):992–997, 1995. DOI: 10.1103/PhysRevA.51.992.

[10] M. Brooks. Quantum computers: what are they good for? *Nature*, 617(7962):S1–S3, 2023. DOI: 10.1038/d41586-023-01692-9.

[11] Google. Google Quantum AI. URL: https://quantumai.google/ (visited on 12/01/2025).

[12] IBM. IBM Quantum Computing. URL: https://www.ibm.com/quantum/ (visited on 12/01/2025).

[13] A. Perdomo-Ortiz et al. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific Reports*, 2(1):571, 2012. ISSN: 2045-2322. DOI: 10.1038/srep00571.

[14] F. Phillipson and H. S. Bhatia. Portfolio Optimisation Using the D-Wave Quantum Annealer. In M. Paszynski et al., editors, *Computational Science – ICCS 2021*, pages 45–59, 2021. ISBN: 978-3-030-77980-1. DOI: 10.1007/978-3-030-77980-1_4.

[15] D. Inoue et al. Traffic signal optimization on a square lattice with quantum annealing. *Scientific Reports*, 11(1):3303, 2021. ISSN: 2045-2322. DOI: 10.1038/s41598-021-82740-0.

[16] L. F. Menabrea and A. Lovelace. Sketch of The Analytical Engine. *Scientific Memoirs*, 3:666–731, 1843. URL: https://www.fourmilab.ch/babbage/sketch.html (visited on 07/04/2025).

[17] L. A. Hiller and L. M. Isaacson. *Experimental Music: Composition With an Electronic Computer*. Greenwood Press, 1959. ISBN: 978-0-313-22158-3.

[18] G. Papadopoulos and G. Wiggins. AI methods for algorithmic composition: a survey, a critical view and future prospects. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, pages 110–117. AISB, 1999.

[19] E. R. Miranda and B. N. Siegelwax. Teaching Qubits to Sing: Mission Impossible? 2022. DOI: 10.48550/arXiv.2207.08225. arXiv: 2207.08225 [quant-ph].

[20] A. Arya et al. Music Composition Using Quantum Annealing. 2022. DOI: 10.48550/arXiv.2201.10557. arXiv: 2201.10557 [quant-ph].

[21] E. R. Miranda. Quantum Computer: Hello, Music! 2020. DOI: 10.48550/arXiv.2006.13849. arXiv: 2006.13849 [quant-ph].

[22] E. R. Miranda et al. A Quantum Natural Language Processing Approach to Musical Intelligence. 2021. DOI: 10.48550/arXiv.2111.06741. arXiv: 2111.06741 [quant-ph].

[23] M. Born and V. Fock. Beweis des Adiabatensatzes [Proof of the Adiabatic Theorem]. *Zeitschrift für Physik*, 51(3):165–180, 1928. ISSN: 0044-3328. DOI: 10.1007/BF01343193.

[24] J. J. Sakurai and J. Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 3rd edition, 2020.

[25] E. Farhi et al. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292(5516):472–475, 2001. DOI: 10.1126/science.1057726.

[26] A. Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2, 2014. ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005.

[27] S. Boixo et al. Experimental signature of programmable quantum annealing. *Nature Communications*, 4(2067), 2013. ISSN: 2041-1723. DOI: 10.1038/ncomms3067.

[28] D-Wave. Minor-Embedding. URL: https://docs.dwavequantum.com/en/latest/quantum_research/embedding_intro.html (visited on 07/04/2025).

[29] D-Wave. Minor-Embedding: Best Practices. URL: https://docs.dwavequantum.com/en/latest/quantum_research/embedding_guidance.html (visited on 18/04/2025).

[30] D-Wave. QPU Solver Parameters. URL: https://docs.dwavequantum.com/en/latest/quantum_research/solver_parameters.html (visited on 18/04/2025).

[31] V. Bapst et al. The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective. *Physics Reports*, 523(3):127–205, 2013. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2012.10.002.

[32] M. W. Johnson et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011. ISSN: 1476-4687. DOI: 10.1038/nature10012.

[33] K. Finley. Quantum Computing Is Real, and D-Wave Just Open-Sourced It. *Wired*, 2017. ISSN: 1059-1028. URL: https://www.wired.com/2017/01/d-wave-turns-open-source-democratize-quantum-computing/ (visited on 17/04/2025).

[34] Next-Generation Topology of D-Wave Quantum Processors. D-Wave Technical Report Series 14-1026A-C, D-Wave, 2019. URL: https://www.dwavesys.com/resources/publications.

[35] D-Wave. The Leap Quantum Cloud Service. URL: https://www.dwavequantum.com/solutions-and-products/cloud-platform/ (visited on 18/04/2025).

[36] N. Science. Golden Record Sounds and Music. URL: https://science.nasa.gov/mission/voyager/golden-record-contents/sounds/ (visited on 17/04/2025).

[37] G. Nierhaus. *Algorithmic composition: paradigms of automated music generation*. Springer, 2009. ISBN: 978-3-211-75539-6.

[38] W. S. Moses and E. D. Demaine. Computational Complexity of Arranging Music. 2016. DOI: 10.48550/arXiv.1607.04220. arXiv: 1607.04220 [quant-ph].

[39] E. Cambouropoulos. The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing. *International Computer Music Association*, 2011. ISSN: 2223-3881.

[40] J.-L. Huang et al. Towards an automatic music arrangement framework using score reduction. *ACM Trans. Multimedia Comput. Commun. Appl.*, 8(1):8:1–8:23, 2012. ISSN: 1551-6857. DOI: 10.1145/2071396.2071404.

[41] S.-C. Chiu et al. Automatic System for the Arrangement of Piano Reductions. In *2009 11th IEEE International Symposium on Multimedia*, pages 459–464, 2009. DOI: 10.1109/ISM.2009.105.

[42] Y. Li et al. Automatic Piano Reduction of Orchestral Music Based on Musical Entropy. In *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5, 2019. DOI: 10.1109/CISS.2019.8693036.

[43] B. Owsinski. *The Mixing Engineer's Handbook*. Bobby Owsinski Media Group, 2017. ISBN: 978-0-9985033-4-9.

[44] D-Wave. dwave-samplers. URL: https://docs.dwavequantum.com/en/latest/ocean/api_ref_samplers/ (visited on 18/04/2025).

[45] S. Kirkpatrick et al. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.220.4598.671.

[46] T. Albash and D. A. Lidar. Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing. *Physical Review X*, 8(3), 2018. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.8.031016.

[47] Zephyr Topology of D-Wave Quantum Processors. D-Wave Technical Report Series 14-1056A-A, D-Wave, 2021. URL: https://www.dwavesys.com/resources/publications.

[48] D-Wave. Decomposing Large Problems. URL: https://docs.dwavesys.com/docs/latest/handbook_decomposing.html (visited on 18/04/2025).

[49] A. Montanez-Barrera et al. Unbalanced penalization: A new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms. *Quantum Science and Technology*, 9(2), 2024. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ad35e4. arXiv: 2211.13914 [quant-ph].

[50] M. Khezri et al. Customized Quantum Annealing Schedules. *Physical Review Applied*, 17(4), 2022. DOI: 10.1103/PhysRevApplied.17.044005.

[51] Z. G. Izquierdo et al. Advantage of pausing: parameter setting for quantum annealers. *Physical Review Applied*, 18(5), 2022. ISSN: 2331-7019. DOI: 10.1103/PhysRevApplied.18.054056. arXiv: 2205.12936 [quant-ph].

[52] Reverse Annealing for Local Refinement of Solutions. D-Wave White Paper Series 14-1018A-A, D-Wave, 2017. URL: https://www.dwavesys.com/resources/publications.

[53] E. Nakamura and K. Yoshii. Statistical piano reduction controlling performance difficulty. *APSIPA Transactions on Signal and Information Processing*, 7, 2018. ISSN: 2048-7703. DOI: 10.1017/ATSIP.2018.18.

[54] M. Pearce and G. A. Wiggins. Towards A Framework for the Evaluation of Machine Compositions. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Sciences*, pages 22–32, 2001. URL: www.doc.gold.ac.uk/~mas02gw/papers/AISB01.pdf.

A Code overview

All code in this study was written in the Python programming language. The most notable libraries used were those included in `dwave-ocean-sdk`[9] for interface with the QPU and `music21`[10] for the manipulation and visualisation of music.

MusicXML[11] was chosen for the digital representation of music. A variant of the established XML (extensible markup language) format, this format focuses on the interactive representation of standard sheet music, describing not just the note content but the structure of the score as well, such as the arrangement of notes into bars and parts. It is widely supported by music notation software, allowing translation of music to graphic (PDF, PNG) and audio (MIDI, MP3) formats, as well as Python libraries (such as `music21`) that provide extensive resources for manipulating such files.

The full repository of code used in this study can be found at

https://github.com/kirbyzx/quantum-arrangement.

B Scores

The following are excerpts of the original scores used for testing this framework, alongside their associated arrangement instruments, problem graphs, and arrangement scores. All original files provided in the public domain. Audio examples can be found in the code repository.

[9] https://docs.dwavequantum.com/en/latest/ocean
[10] https://www.music21.org/music21docs
[11] https://www.musicxml.com

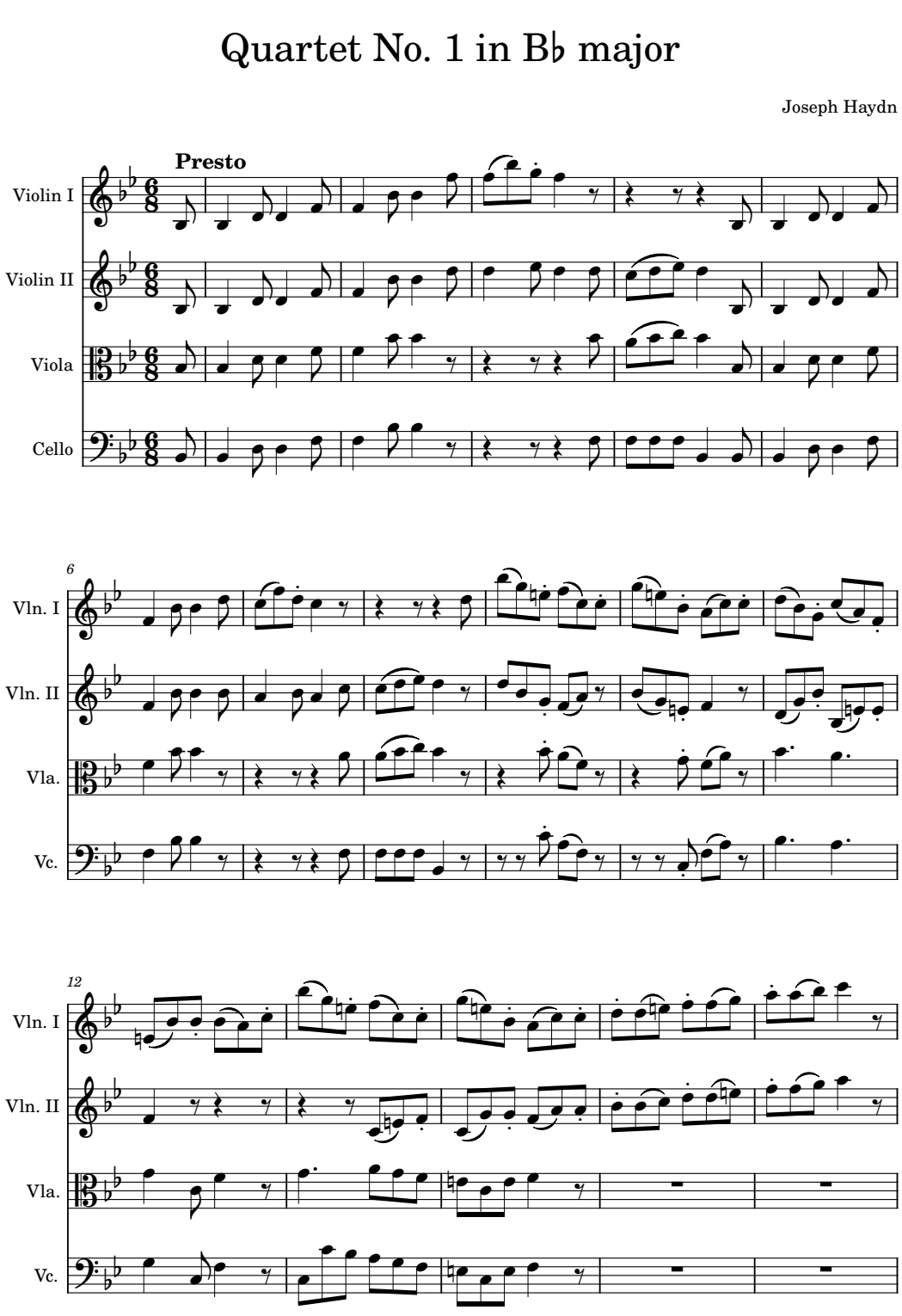


Figure B1: Quartet No. 1 in B-flat major, Op. 1, by Joseph Haydn, bars 1–16.

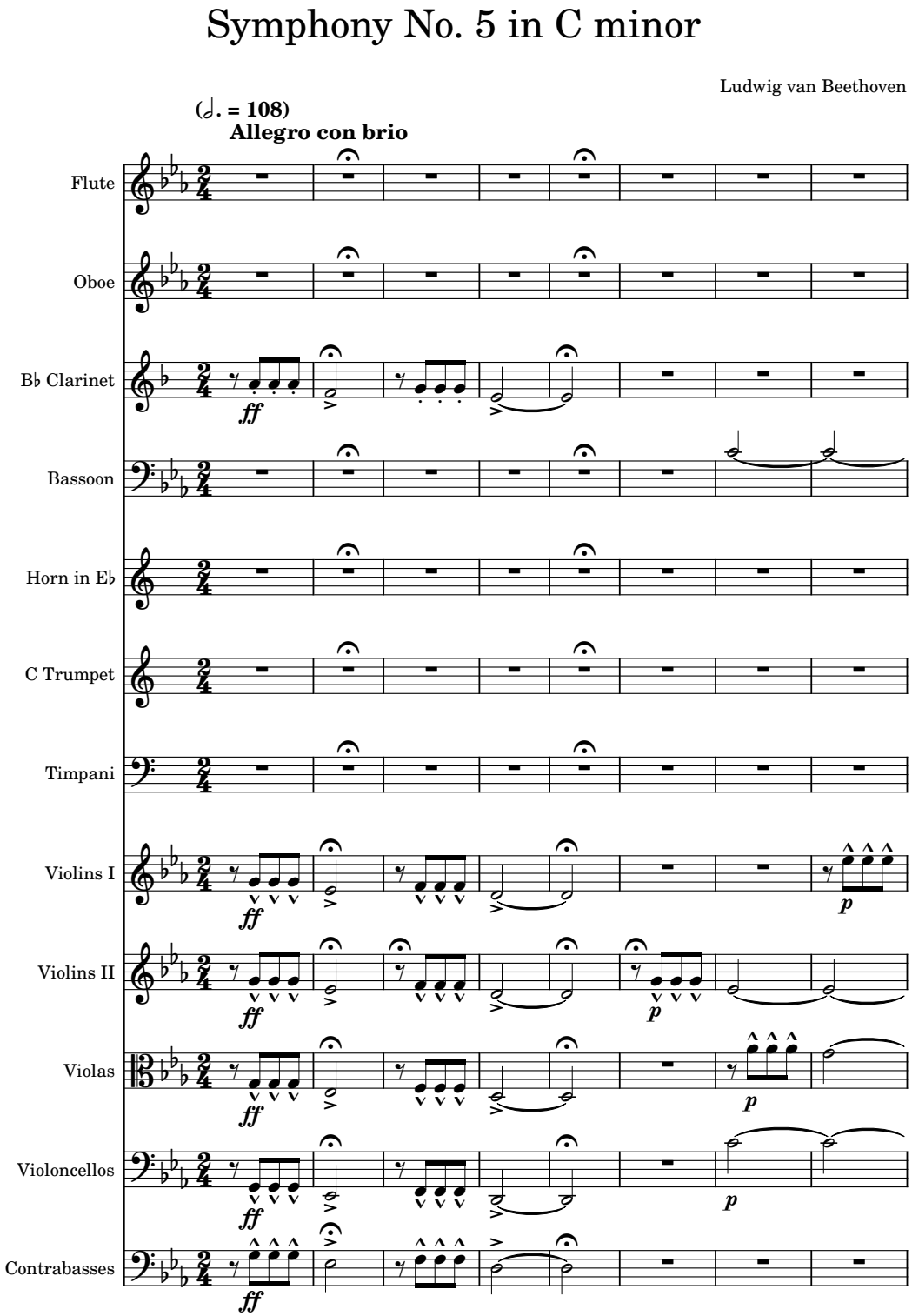


Figure B2: Symphony No. 5 in C minor, Op. 67, by Ludwig van Beethoven, bars 1–8.

Scientific summary for a general audience

The arrangement of music by hand is usually a difficult and time-consuming process, requiring a deep understanding of musical theory and structure. This study aims to automate this process via quantum computing, a technique that relies on the use of qubits, which can exist in a superposition of states. A music score can be split up into a sequence of phrases by looking at how much adjacent notes differ from each other, and turned into a graph representation with nodes and edges, where each node is a phrase, and edges between nodes mean they overlap. This graph can then be sent to a quantum computer in order to select nodes according to a set of rules that determine the properties of the arrangement. Once the nodes have been selected, the corresponding phrases can be reconstructed to create the final score. Here, an excerpt of Beethoven's String Quartet No. 10 is reduced to a single part, suitable for a solo instrument.