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. This method is tested on an excerpt of Beethoven's String Quartet No. 10 in E-flat major; in this case, the optimisation process is successful, and selects compatible phrases that produce a suitable arrangement.

Durham University

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 [CITE]. 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. With the age of quantum computing came the possibility to solve complex problems usually intractable to even the most powerful classical computers. 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 **googleibm**. 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, which often involves brute-force methods. Adiabatic quantum computers have been used to find solutions to a variety of optimisation problems, such as protein folding **perdomo-ortiz_protein_2012**, financial portfolios **phillipson_portfolio_2021**, and traffic flow **inoue_traffic_2021**.

At time of writing, quantum computing is still a nascient field, with quantum computers remaining impractical for real-world applications. In theory, larger and more advanced quantum computers would be able to solve problems in much shorter timeframes than classical computers, an advantage dubbed "quantum supremacy". The achievement of this would trigger an upheaval of many common computer systems, most importantly encryption and simulation [CITE]. A recent study has claimed to demonstrate this advantage using adiabatic quantum computing to analyse the structure of magnetic materials [CITE], although this is a specialised application and far from demonstrating an overall "supremacy" over classical methods.

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] [CITE], with Ada Lovelace, considered the first computer programmer, noting that "the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent". The first computer composition would not be produced until nearly a century later [CITE], and since then various approaches to computer music have been taken, ranging from fractals [CITE], Markov chains [CITE], and evolutionary models [CITE]. 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 [CITE], develop harmonies [CITE], create synthesisers [CITE], produce variations of original scores [CITE], and create intelligent musical systems via quantum natural language processing (QNLP), using a mixture of gate-based and adiabatic methods. However, 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

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

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 version 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 a detailed framework for combining the two, and apply this method to two musical scores; finally, we consider conclusions and possible future work.

2 Quantum annealing

Before we consider quantum annealing, we must first discuss adiabatic quantum computing (AQC), from which it draws its main principle. AQC is a computing technique that works under the adiabatic theorem.

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.* ***born_beweis_1928***

During a truly adiabatic process, there is a perfect conservation of entropy within the system, so that system must remain in its energy eigenstate. This theorem can also be seen as a consequence of the energy-time uncertainty relation, which can be expressed as

$$\Delta E\, T \geq \frac{\hbar}{2} \tag{1}$$

where here we define ΔE as the uncertainty in energy eigenstate and T the time interval over which the system is evolved. As $T \to \infty$ we allow $\Delta E \to 0$, meaning the exact energy eigenstate of the system can be known. A typical evolution of the Hamiltonian of the system can be expressed as

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_p \tag{2}$$

where we are evolving from an initial Hamiltonian H_0 to a final Hamiltonian H_p. 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). 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 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, but follows a probability distribution. 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.

[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.

The main use of quantum annealing is 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 prepared such that its ground state encodes the solution to the optimisation problem, then as long as the initial Hamiltonian is prepared in the ground state, the solution is likely to be given 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 nondeterministic). Problems like these have large solution spaces with many local minima, which cannot be solved quickly. A common example is the travelling salesman problem: a salesman must visit a set of cities exactly once and return home, whilst minimising the distance travelled [CHECK THIS]. As the number of cities increases, the number of possible routes the salesman could take grows exponentially, with a classical algorithm having to consider many more possible options. It is these sorts of optimisation problems that quantum annealers excel at solving.

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) [CHECK THIS]. 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 the Ising model, with the discrete variables now *qubits*—binary variables like their classical counterparts, but existing in a superposition of the two states until measurement. The corresponding ground state is prepared with the Hamiltonian

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

which is an equal superposition of all possible states in the eigenbasis of H_p [CHECK THIS].

Another way of expressing problems is via the QUBO (quadratic unconstrained binary optimisation) model. A QUBO model takes the form of a function $f(x)$ to be minimised, and takes the form

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

where $x \in [0, 1]$ 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 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

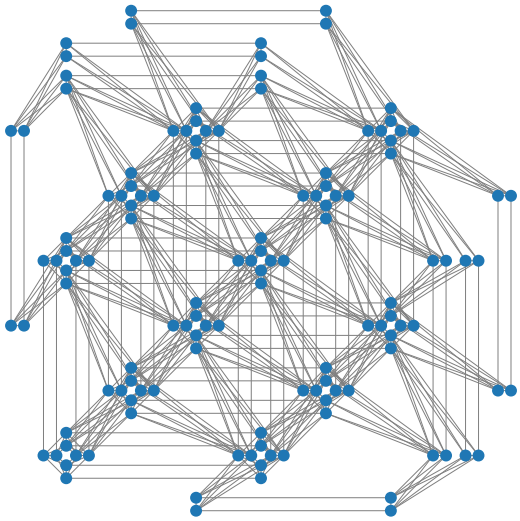

Figure 1: A graph of 144 qubits in a D-Wave QPU, using their Pegasus topology. Qubits are represented by vertices, and couplers by edges.

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 lattice of physical qubits connected via couplers. The qubit biases and coupling values are influenced by electromagnetic fields, allowing the mapping of models to physical qubits, with linear terms as qubits 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, 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 the solution.

Often, the topology of a QPU doesn't 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 allowed 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. The chain strength determines how strongly the chain is coupled, enforcing that 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 switch 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.

QPU annealers (also known as solvers) have many other parameters that can be tuned, influ-

encing exactly how the annealing process is carried out and significantly affecting the quality of solutions [CITE SOLVER WEBSITE]. Aside from chain strength, others that this study considers are anneal time per sample and number of reads. Both these parameters affect the total QPU access 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. For a problem with N qubits there are 2^N energy levels, with the relative size of spectral gaps (differences between energy levels) decreasing with N [ASK ABOUT THIS]. As seen in Equation (1), smaller spectral gaps (ΔE) require longer anneal times (T) to prevent excitation to higher energy states, such that the uncertainty in energy eigenstate is not large enough to eclipse this gap. (minimum energy gap?) The exact time evolution during this interval can also be customised to control results further **khezri_customized_2022**, although this is beyond the scope of this study.

A benefit of shortened anneal times is that the system can be evolved and measured (sampled) many times, increasing the likelihood that a lower-energy solution from the distribution will be found. An evolution sampled many times creates a distribution in solution energy: for a simple QUBO model with a sparse matrix Q, this distribution may be biased towards lower energies therefore needing fewer samples; for a complicated matrix with many entries, the sampled distribution tends towards a Gaussian [RANDOM MATRIX LIT], with many more reads necessary to be likely to sample from the tail end. The density of a problem matrix can be found from a QUBO model of the form of Equation (6) as

$$\rho = \frac{V + E}{V^2} \tag{8}$$

where we have defined a new graph $Q = (V, E)$, with V as the number of linear coefficients and E the number of quadratic coefficients. $\rho = 1$ indicates a fully-connected problem, the worst-case scenario.

Many companies provide both personal and commercial access to gate-based quantum computers, however, few develop the hardware necessary for quantum annealing. D-Wave Systems was the first to realise a true quantum annealer, and have since developed and released their technology to the wider scientific community. All problems in this study were submitted to D-Wave's Advantage System 4.1 solver, which boasts a topology of over qubits each with a degree of. held at cryogenic temperatures and low-magnetic field environments to prevent unwanted fluctuations.

D-Wave Systems operates the first and only quantum annealer for commercial access, introduced in 2011 . All problems in this paper are run on the D-Wave Advantage quantum system, which uses their Pegasus architecture, an example of which can be seen in Figure 1. A full Advantage QPU contains over 5000 qubits, each with a maximum degree of 15. Interaction with the D-Wave QPUs is handled through their Leap quantum cloud service, 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 translate problems into binary optimisation and create problem models for the QPU to solve.

Quantum annealing has already been applied to a widge array of optimisation problems, ranging from financial portfolios, protein folding, and traffic flow. This study explores a creative novel application of this technique, music arrangement.

!!!Nick Chansellor paper, why quantum annealing?

3 Music arrangement

The first evidence of humans writing music is theorised to have originated as early as 2000BC [CITE]; since then, it has become a hallmark of the human experience, so much so that the *Voyager I* probe carries over 50 tracks of music from across the world as a message to potential extraterrestrial life [CITE]. Whilst some musicians would compose entirely new music, others would build upon and adapt previously composed pieces for practical or artistic reasons, whether that be in terms of instrumentation, medium, or style, to create an *arrangement*. One of the most famous examples of this is *Pictures at an Exhibition*, a piano suite written by Modest Mussorgsky, but more commonly known for its orchestral arrangement by Maurice Ravel. Traditionally, the arrangement of music is a complex process that requires a deep understanding of musical theory, structure, and style. Composers use their creativity and intuition 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 this can be seen in the *Musikalisches Würfelspiel* ("musical dice game") system popular in the 18th century. The roll of dice would determine the order of pre-composed musical phrases, allowing for the creation of new music 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. However, these methods are limited by the complexity of the problem and the need for extensive training data.

Music arrangement refers to the art of rewriting a musical score in a new setting, style, or structure, whilst still maintaining the essence of the original. A classical example of this is Mussorgsky's *Pictures at an Exhibition*, which is well known for its orchestral arrangement by Ravel but was originally a piano solo. A more contemporary example would be ???'s rendition of *The Star-Spangled Banner* at the ??? Super Bowl, which caused much controversy due to its laid-back style during a time of conflict in Cuba? 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.

Traditionally, it's a time-consuming blah blah blah. However, in this study we aim to formulate music arrangement as an optimisation problem, which can be solved like any other via quantum annealing. It has been shown that music arrangement can be reduced to an NP-hard problem **moses_computational_2016**, by considering the special case of music *reduction*. Reduction is a form of arrangement whereby the number of instrument parts only gets smaller, and we define here subject to the following conditions:

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

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

Condition 3 (Monophonic)**.** Each part in the arrangement can only contain music from a single part in the original at any time.

Holistically, reduction aims to fit as much original music into the parts of the arrangement, whilst still being playable. Notably, these constraints forgo the need to compose new music, a very subjective endeavour that is not handled well by optimisation.

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 European Classical-era (?) small ensemble and orchestral music, written in standard Western notation.

!!!mention somewhere score taxonomy (instruments, parts, polyphony, etc.)

4 Proposed framework

Previous approaches to the automatic arrangement of music have relied on classical methods such as neural networks or recursive classical algorithms 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, a new approach to the automatic arrangement of music is proposed by applying quantum annealing.

: given a set of musical phrases, each phrase can be assigned a binary variable and formulated into a Boolean satisfiability (SAT) problem, with variables assigned to clauses according to their compatibility. A valid solution would be an assignment of values such that the Boolean expression is satisfied, corresponding to the phrases included in the final arrangement . Phrases are chosen as the smallest unit of music instead of notes to best preserve important melodic lines and harmonies, which may sound dissonant or confusing if split up.

It has been shown that the arrangement of music can be formulated as a combinatorial optimisation problem **moses_computational_2016** We propose a new framework for the arrangement of music via quantum annealing, 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

The beginning of any arrangement process is an original musical score. However, before reduction can begin, the score must undergo preprocessing to ensure the produced arrangement will fulfil the conditions outlined in Section 3. In particular, Condition 3 demands that all phrases must be strictly monophonic. Some instruments (e.g. strings) are capable of playing multiple notes at a time, 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 simultaneosly. 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 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 important melodic lines and harmonies, we maintain the essence and familiarity of the original score, which is the core principle of arrangement as outlined in Section 3.

The approach taken in this study is the local boundary detection model (LBDM) **cambouropoulos_lbdm_2011**, 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 [CITE?]. The strength of a boundary on a particular note is calculated over two of its parameters: pitch, and inter-onset interval (IOI), the time until the next note. This is chosen over note duration as it is often more noticeable to the listener [CITE] (pithy quote about hearing the gaps between notes?). The boundary strength S for a particular parameter x at note i is given by

$$S_{x,i} = x_i \cdot (r_{i-1,i} + r_{i,i+1}) \tag{9}$$

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

$$r_{i,i+1} = \frac{|x_i - x_{i+1}|}{x_i + x_{i+1}}\,. \tag{10}$$

In this way, a note with a high boundary strength would signal the start of a new musical phrase. The set of strengths for each parameter is normalised to the range $[0, 1]$ (CHECK NOTATION) via

$$S'_{x,i} = \frac{S_{x.i} - \min(S_x)}{\max(S_x) - \min(S_x)} \tag{11}$$

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 signals 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 a boundary and the one following as the start and end points. 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 many pre-existing NP problems that can be solved in this way, including Boolean satisfiability

and set theory **lucas_ising_2014**. 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.[3] This study uses a problem in the category of graph colouring known as proper vertex 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 vertices that overlap (i.e. played different parts 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 conditions outlined previously: 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 simultaneously; Condition 3 is met by adjacent colours constraint, meaning a single part cannot play two overlapping phrases simultaneously.

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

$$x_{v,i} = \begin{cases} 1 & \text{if vertex } v \text{ is colour } i \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

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

$$f(x) = A \sum_{v \in V} \left(s_v - \sum_{i=1}^{n} x_{v,i} \right)^2 + B \sum_{(u,v) \in E} \sum_{i=1}^{n} x_{u,i} x_{v,i} \tag{13}$$

where A, B are the Lagrange parameters, and we have introduced the slack variables $s_v \in \{0, 1\}$ (CHECK NOTATION).[4] 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. The nature of a reduction implies that not all phrases will be played, so the inclusion of the slack variables (specifically when $s_x = 0$) 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 B for each pair of identically-coloured vertices connected by an edge. When this function is minimised (strictly zero, in this case), 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 2d.

4.4 Musical entropy

A valid arrangement of phrases does not good music make. To ensure the produced arrangement is as musically interesting as possible, each vertex of the graph can be weighted to introduce

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

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

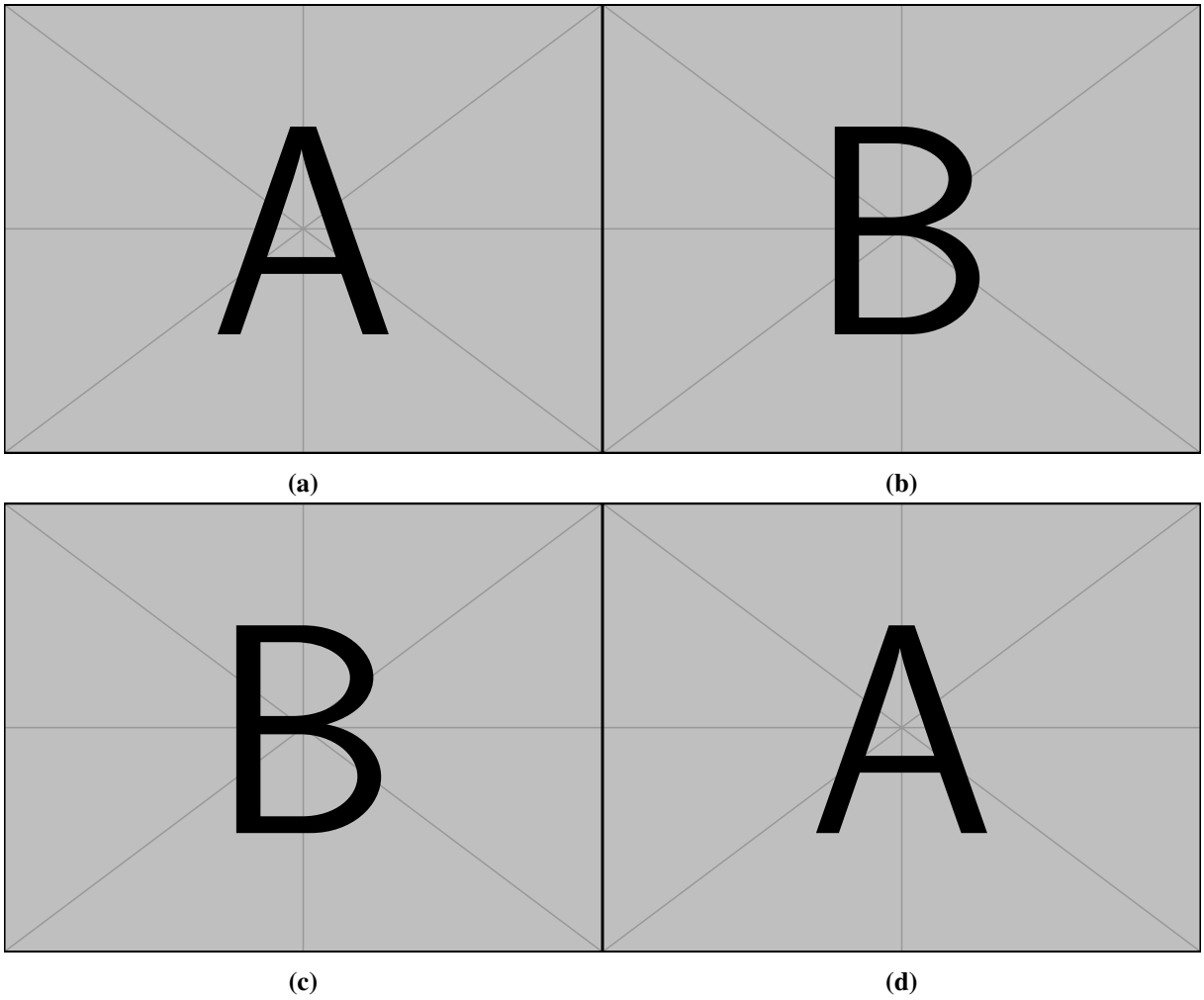

Figure 2: A toy example demonstrating the proposed framework. (a) An example score of three parts, with phrases identified by the LBDM coloured. (b) A graph of the score, with the colour of each node corresponding to its phrase, and edges connecting overlapping phrases. (c) One possible solution of proper vertex colouring, with $n = 2$. (d) The score representation of the final arrangement.

a bias according to its phrase's 'musicality'. The idea of a musical 'utility' has been used before with classical algorithms **huang_towards_2012** and to assess playability rules [PIANO REDUCTION CITATION]. Here we quantify this by calculating the *musical entropy* of a phrase **li_entropy_2019**.

Entropy can be seen as how much information a variable contains; in this context, a phrase with a higher entropy would indicate a higher level of variation and complexity, increasing the likelihood of it being more 'musical'. Maximisation of phrase entropy would then 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

$$H(X) := -\sum_i P(x_i) \log_2 P(x_i) \tag{14}$$

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

$$P(x_i) = \frac{n_i}{N} \tag{15}$$

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 considering the duration of the note instead. This is also calculated using Equation (15), 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.

We can modify Equation (6) to include vertex weights by adding an objective term that looks like

$$f_C(x) = -C \sum_{v \in V} \sum_{i=1}^{n} W_v x_{v,i} \tag{16}$$

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 this function results in the maximisation of the objective (the sum of the total weights).

Weights can also be added to edges, to bias the selection of pairs of vertexes. 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

$$f_D(x) = -D \sum_{(u,v) \in E} W_{uv} \sum_{i=1}^{n} \sum i < j^n x_{u,i} x_{v,j} \tag{17}$$

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 **owsinski_mixing_2017**. Here we consider only three: lead, often including melodic/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 bass. 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 a lead in the arrangement should contain more high-entropy phrases than a cello part assigned as foundation. The corresponding objective term takes the form

$$f_E(x) = -E \sum_{v \in V} \sum_{i=1}^{n} \theta(W_v - W_{\text{th},i}) x_{v,i} \tag{18}$$

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.

Together, the final QUBO model is

$$\begin{aligned} f(x) &= f_{A,B} + f_C + f_D + f_E \\ &= A \sum_{v \in V} \left(s_v - \sum_{i=1}^{n} x_{v,i} \right)^2 + B \sum_{(u,v) \in E} \sum_{i=1}^{n} x_{u,i} x_{v,i} \\ &\quad - C \sum_{v \in V} \sum_{i=1}^{n} W_v x_{v,i} - D \sum_{(u,v) \in E} W_{uv} \sum_{i=1}^{n} \sum_{i<j}^{n} x_{u,i} x_{v,j} - E \sum_{v \in V} \sum_{i=1}^{n} \theta(W_v - W_{\text{th},i}) x_{v,i} \end{aligned} \tag{19}$$

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 (19) does indeed fulfil the constraints necessary for a valid arrangement as outlined before, the Lagrange parameters must be tuned. It is possible for this to be done *a priori* [CITE], however, with five parameters, 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 (19) 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 across parts 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 lower the energy maximally whilst breaking a constraint. To ensure that this results in an overall increase in energy, and is therefore less likely, we can derive the inequality

$$A + B > C \max(W) + D \max(W) + E \tag{20}$$

where $\max(W)$ denotes the largest entry of the weight matrix. To allow a similar analysis across problems with different weights, 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

$$A_m + B_m > 2 + E_m \,. \tag{21}$$

In this study we set $E_m = 1$ as bias towards part types is preferred but not essential. 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 constraints, whilst not being too large as to overwhelm the other terms in the QUBO equation [CITE] and affect the quality of solutions.

4.6 Solution interpretation

Once the problem has been submitted to a quantum annealer and a sample set returned, we can finally construct the arrangement. Of the distribution of 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.

This is the point where music post-processing can be applied. In this study, phrases are shifted up or down octaves depending on the pitch range of the target instrument, to ensure playability whilst minimally affecting the essence of the original.

5 Results

The outlined framework was applied to two scores: Quartet No. 1 in B-flat major, Op. 1, by Joseph Haydn, and Symphony No. 5 in C minor, Op. 67, I. Allegro con brio, by Ludwig van Beethoven. Haydn Op. 1 is of the 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 in the QPU. Beethoven Op. 67 is of the 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. Notably, the piece is also very well-known meaning the constructed arrangements would be more familiar. Original scores, and their reduction counterparts, can be found in Appendix B.

Music can be represented by many digital formats, each with their own benefits—in this study we choose to use MusicXML [CITE], a variant of the well-establish XML (extensible markup language) format. This format focuses on the interactive representation of standard sheet music, describing musical notes, rests, and other symbols, as well as the structure of the music, such as the arrangement of notes into bars, parts, and scores. 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 that provide extensive resources for manipulating, translating, and creating MusicXML files, allowing scores to be broken down and reconstructed

as fit.[5] Although not as prevalent as other file formats, there exist a number of online libraries and databases of MusicXML files provided in the public domain.

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. 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. The entropy of each phrase was calculated using Equation (14), to be used as the corresponding vertex weight and to calculate the edge weights, and the QUBO model was then calculated using Equation (19), which could then be sent to the QPU to be solved. 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 (as this was the greatest problem size the QPU allowed), but now as a string quartet (Violin I, Violin II, Viola, Violoncello).

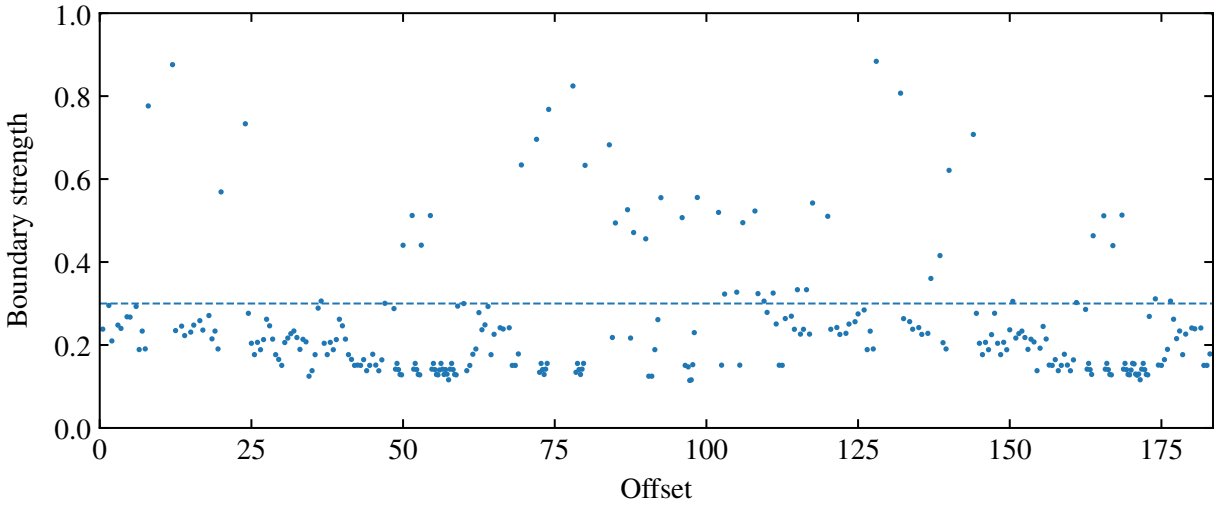

Figure 3: Calculated boundary strengths for the Violin I part of Haydn Op. 1, using pitch and IOI weightings of 0.33 and 0.66 respectively. The threshold of 0.3 has been denoted by the dashed line, resulting in 51 identified boundaries.

The tuning of the Lagrange parameters A_m and B_m can be seen in Figure 4. These two parameters were varied together as the complexity of the QUBO equation makes it hard to predict how the terms interact. In terms of reducing the number of broken constraints, the edge multiplier (B_m) can be seen to have the most importance, although an increase in the vertex multiplier (A_m) is still requried to reduce this number to exactly zero in order to produce a valid solution. The parameters used for subsequent models were $A_m = 6$ and $B_m = 6$, which can be seen to fulfil Equation (21) as expected. Each constraint parameter gives an increase in energy roughly double the potential energy decrease for a broken constraint, a value which has been shown before (???) [CITE LIT]. This does not guarantee that no constraints are broken in the lowest-energy solutions, however, non-zero broken constraints will be rare. From this point onwards, only the lowest-energy solutions with zero broken constraints are considered.

An example of the distribution of a returned sample set can be seen in Figure 5. The most optimal solutions lie in the leftmost portion of the distribution, corresponding to the lowest energies. The Gaussian nature of this distribution changed very little throughout parameter

[5] An overview of the code resources used in this study can be found in Appendix A.

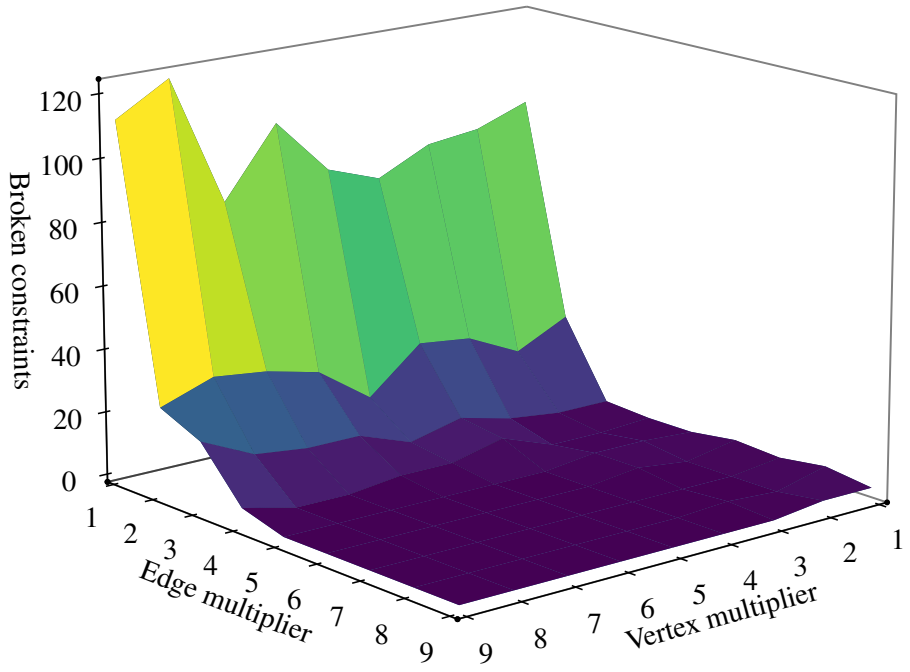

Figure 4: Parametric plot from tuning the Lagrange parameters A_m (vertex multiplier) and B_m (edge multiplier). It can be seen that increasing the edge multiplier results in the most drastic change in the number of constraints broken. The QUBO equation for each pair of parameters was solved 10 times, and the mean taken.

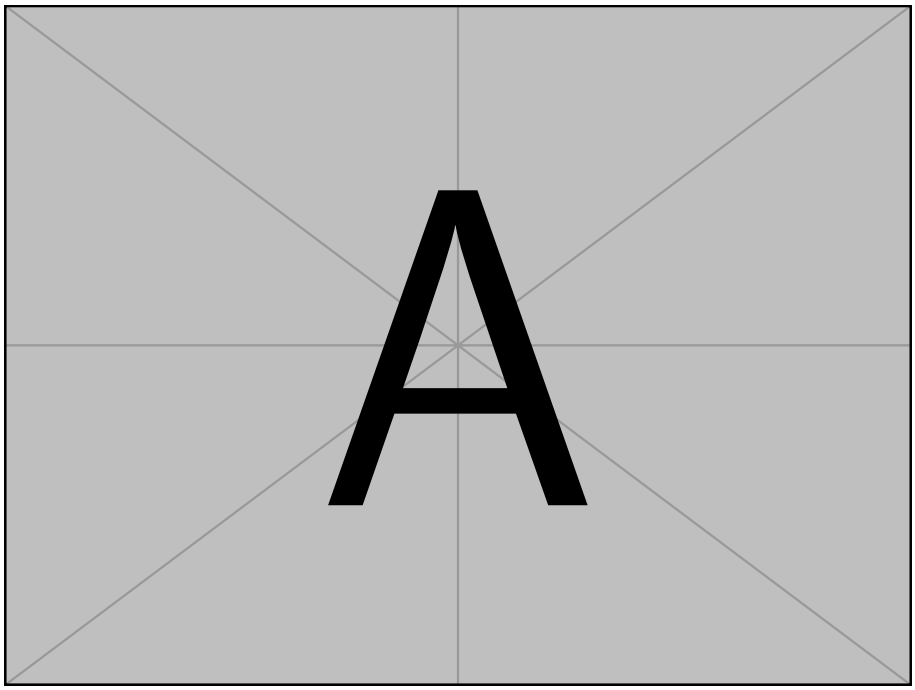

Figure 5: Histogram example of a returned sample set.

configuration process, hinting at the possibility that the problem models were already complicated enough to resemble random matrices. How sparse is the problem graph? $(nodes + edges)/nodes^2$

Optimisation of both the chain strength and anneal time per sample solver properties can be seen in Figure 6. Whilst usually the aim of increasing the chain strength is to reduce the fraction of broken chains, here this is not too significant as long as the constraints are met. Broken chains 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 the lowest energy solutions can still be valid. As can be seen in Figure 6a, a chain strength greater than 30 reduced the number of broken chains to zero, but the lowest energy of valid solutions was minimised at a lower value of 20. For further problems, a chain strength of 25 was chosen as a compromise between energy and chain break fraction minimisation.

Variation of anneal time can be seen in Figure 6b. This could only be increased to a maximum of 300 µs for a single problem submission due to time limits imposed by the solver. In theory, as $t \to \infty$, the energy of the samples should tend to the ground state energy due to the adiabatic theorem; here the energy seems to decrease exponentially to some asymptotic value. 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 [???]. 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 future problems.

Having tuned the Lagrange parameters and sampler configuration to increase the probability of finding the most optimal solutions, we compare the returned solutions to classical optimisation algorithms solving the same problem. D-Wave provides a selection of classical samplers that can return solutions to QUBO problems, two of which are used in this study. The steepest descent method blah blah, and despite its apparent rudimentary nature is still very widely used

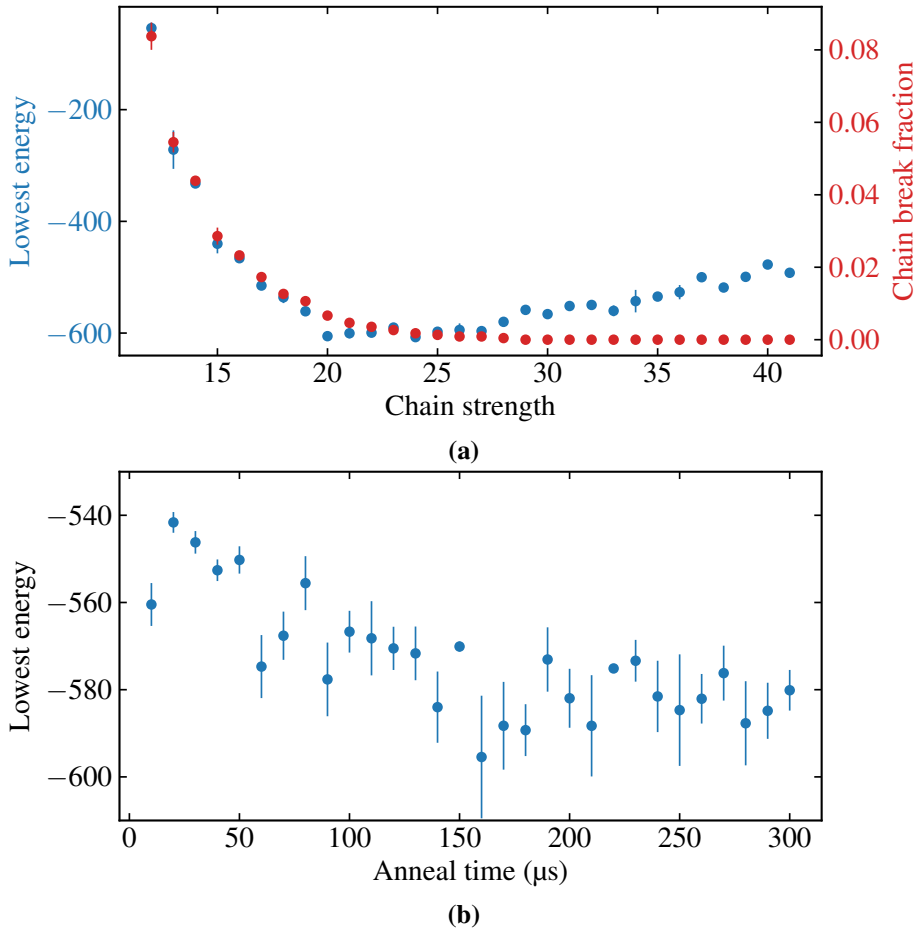

Figure 6: Optimisation of the solver configuration. (a) Variation of the fraction of chains broken and energy with chain strength of the lowest energy samples in a sampleset of 1000 reads, using the default anneal time (20 µs). Each data point was sampled ?? times, with the mean and standard error calculated. (b) Variation of ?? with anneal time per sample of the lowest energy samples in a sampleset of 1000 reads, using the default chain strength (32).

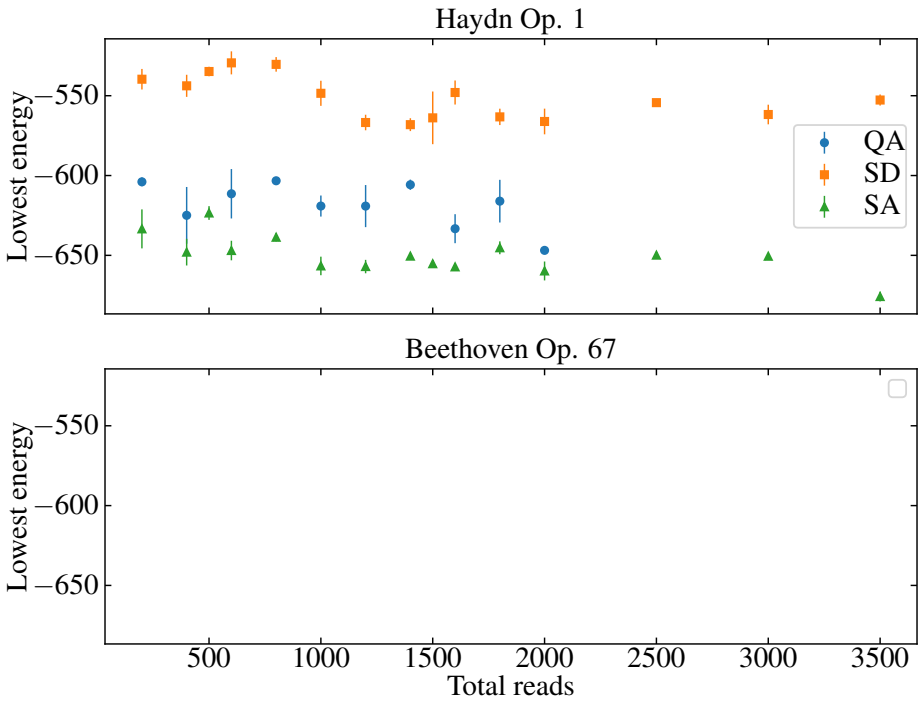

Figure 7: Comparison of quantum annealing (QA) with steepest descent (SD) and simulated annealing (SA) classical solvers, varying the number of reads. Quantum annealing consistently provides lower-energy solutions than steepest descent, although fails to beat simulated annealing.

for solving optimisation problems [CITE]. Simulated annealing, on the other hand, works very similarly to quantum annealing, but instead uses a ”temperature” value to escape local minima. This method gives a more direct comparison of the effectiveness of quantum and classical approaches.

A comparison of the solutions returned by the different solvers for both pieces can be seen in Figure 7. Quantum annealing consistently provides lower-energy solutions than steepest descent, although fails to beat simulated annealing by a small margin. For the larger problem posed by Beethoven Op. 67, ...Why???

1. Score length
2. Number of instruments

!!!comparing time to solution For QA only looking at annealing time, not QPU access time (which includes sampling, readout, programming time) as this is a limitation of the technology rather than the method

6 Conclusions

Although the technology has been improving for several years, at time of writing quantum annealers are still in the development stage. D-Wave’s latest chip, the Advantage 4.1, boasts a topology of over 5000 vertices

QPU chip graph size -¿ affects length of scores/granularity of phrases/number of arrangement instruments !!!is possible to partition graphs to reduce problem sizes (include citation) but is future work

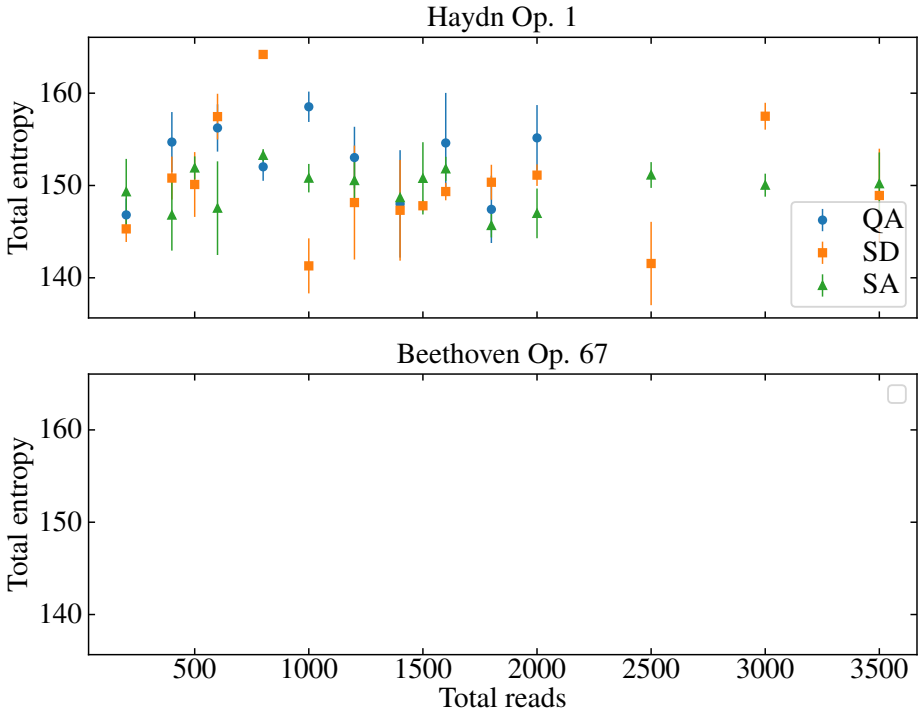

Figure 8: Comparison of quantum annealing (QA) with steepest descent (SD) and simulated annealing (SA) classical solvers, varying the number of reads. Quantum annealing consistently provides lower-energy solutions than steepest descent, although fails to beat simulated annealing.

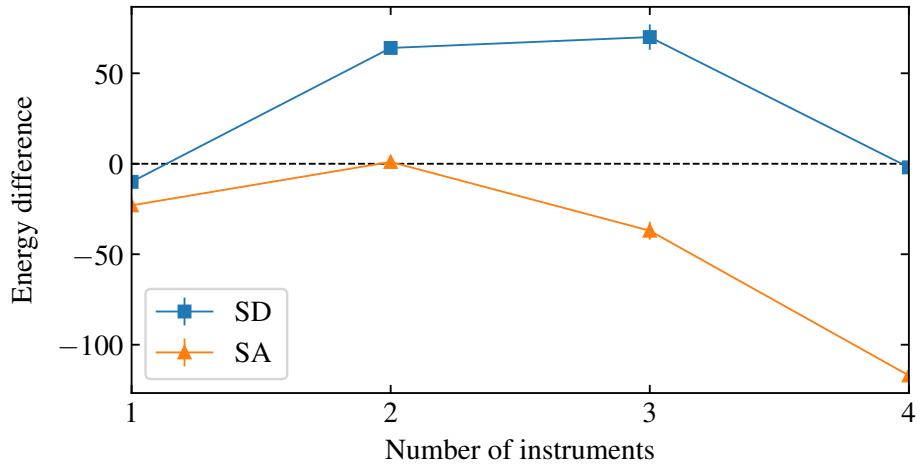

Figure 9: Scaling with number of instruments.

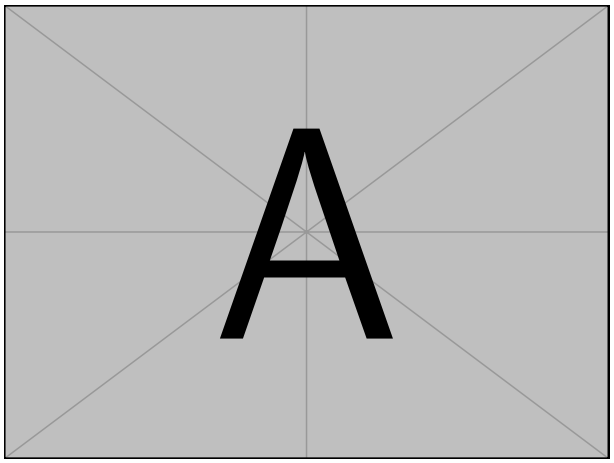

Figure 10: Increase in variables with number of instruments (one for linear terms one for quadratic terms)

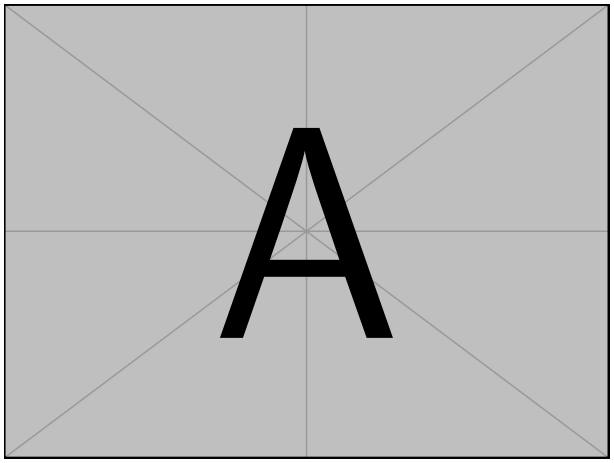

Figure 11: Comparing time to solution with number of instruments

Maximum problem time limit -¿ affects maximum time per anneal which could affect quality of results

Removing chords and voices from scores -¿ loss of information

!!!what is the main limiting factor? Computers aren't big enough yet, but how big would they need to be? Look into new Zephyr topology

!!!future work reverse annealing change in anneal schedule, pause/quench graph partitioning a priori lagrange parameters (Nelder-Mead) unbalanced penalisation vs slack penalisation (would reduce number of variables)

In the case of Beethoven's String Quartet No. 10, this method can be said to be successful in reducing the score to a single monophonic part. A sufficient number of phrases are identified to allow the key melodic lines to be picked out from each part, and the weighted MIS formulation results in a final arrangement that is not musically uninteresting. Here, only one solution was picked arbitrarily; if desired, each of the degenerate lowest energy solutions could be examined individually to isolate the one that is considered most "musical", safe in the knowledge that all of them are valid.

One advantage this method has over classical algorithms is that no training data is needed. Genetic and deep learning approaches require refinement of their models on large datasets, "teaching" what a valid arrangement looks like, which takes considerable time and resources. By using a QPU, these "rules" can effectively be encoded using constraints, such as the penalty term introduced into the QUBO; as long as these constraints are fulfilled, the QPU will always provide a feasible solution, without any knowledge of previous arrangements.

Another benefit is the solve time: for the problem graph shown in Fig. , the D-Wave QPU access time (the time taken to embed and solve the problem) is approximately 150 ms for 1000 reads. Compared to classical methods using a similar number of phrases, the quantum process is faster by at least one order of magnitude . For classical computers, the time taken to solve NP-hard problems increases exponentially with the problem size, so this speed advantage will only improve as the scores become more complex.

Future work would include testing this process on a wide range of pieces from different musical styles. Scores from different time periods (Baroque vs. Romantic, for example) have wildly different structures, and so the effectiveness of the LBDM could vary in identifying suitable musical phrases. The treatment of melodic lines between different ensembles, whether that be a string quartet or symphony orchestra, also varies, and the isolation of important melodic lines may become more difficult.

To that regard, the effect of LBDM parameters (pitch/IOI weighting, threshold) on phrase identification could be studied. Ideally, musical phrases should be fairly short and similar in length, to give the MIS 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. Other boundary detection methods could also be tested, with their resulting identified phrases compared qualitatively.

Currently, no regard is taken to the exact instrumentation of the final arrangement. Different instruments have varying ranges of pitches that can physically be played; here, phrases have been chosen from across all the instrument parts (as seen in Fig.), however, this does not mean that the final arrangement can be played by all the instruments as some notes may be unfeasible. Care needs to be taken to check the ambitus (pitch range) of chosen phrases and ensure that it

falls within the desired instrument's range; if not, phrases can be transposed by octaves up or down without altering the melodies significantly.

The MIS formulation works well for reducing a score into a single polyphonic part, as the inclusion of a phrase within the final arrangement lends naturally to the binary variables used. However, a more useful application would be the ability to reduce a score to any fewer number of parts, whether that be multiple monophonic instruments (e.g. string quartet) or a polyphonic instrument (e.g. piano). To achieve this, the QUBO would need to be altered to allow some edges into the final subset. One suitable formulation would be a graph colouring problem: colouring the vertices of a graph G with a set of n colours such that no edge connects two vertices of the same colour. In this context, the number of colours would be the number of desired parts—after solving, each disjoint set of colours would become a monophonic part, the combination of which becomes the final reduction (in this way, the MIS problem can be seen as a colouring problem where $n = 1$).

Although it can be argued that music cannot be objectively "scored", nonetheless, the quality of the produced arrangements needs to 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 . 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, via a series of blind trials. If the participants fail to identify the computer compositions from the human ones more often than random chance, then it can be said that the generated arrangements are of sufficiently good quality.

To conclude, this paper has shown the automatic reduction of music via quantum annealing. By formulating a score into a function to be minimised, the annealing process can identify parts of the original to become the final arrangement. An automatic process of this sort would be useful to a wide calibre of musicians, removing the time and skill barrier to produce such arrangements, whilst still retaining a sufficient level of quality, keeping music accessible to all. Just as science and business do already, it can be hoped that the arts take advantage of promising new technologies as well.

References

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

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

perdomo-ortiz˙protein˙2012 Alejandro Perdomo-Ortiz et al. "Finding low-energy conformations of lattice protein models by quantum annealing". In: *Scientific Reports* 2.1 (Aug. 2012), p. 571. ISSN: 2045-2322. DOI: 10.1038/srep00571. URL: https://www.nature.com/articles/srep00571 (visited on 01/12/2025).

phillipson˙portfolio˙2021 Frank Phillipson and Harshil Singh Bhatia. "Portfolio Optimisation Using the D-Wave Quantum Annealer". In: *Computational Science – ICCS 2021*. Ed. by Maciej Paszynski et al. Cham: Springer International Publishing, 2021, pp. 45–59. ISBN: 978-3-030-77980-1. DOI: 10.1007/978-3-030-77980-1_4.

inoue˙traffic˙2021 Daisuke Inoue et al. "Traffic signal optimization on a square lattice with quantum annealing". In: *Scientific Reports* 11.1 (Feb. 2021), p. 3303. ISSN: 2045-2322.

DOI: 10.1038/s41598-021-82740-0. URL: https://www.nature.com/articles/s41598-021-82740-0 (visited on 01/12/2025).

born˙beweis˙1928 M. Born and V. Fock. "Beweis des Adiabatensatzes". de. In: *Zeitschrift für Physik* 51.3 (Mar. 1928), pp. 165–180. ISSN: 0044-3328. DOI: 10.1007/BF01343193. URL: https://doi.org/10.1007/BF01343193 (visited on 03/01/2025).

khezri˙customized˙2022 Mostafa Khezri et al. "Customized Quantum Annealing Schedules". In: *Physical Review Applied* 17.4 (Apr. 2022), p. 044005. DOI: 10.1103/PhysRevApplied.17.044005. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.17.044005 (visited on 10/11/2024).

moses˙computational˙2016 William S. Moses and Erik D. Demaine. *Computational Complexity of Arranging Music*. arXiv:1607.04220. July 2016. DOI: 10.48550/arXiv.1607.04220. URL: http://arxiv.org/abs/1607.04220 (visited on 11/09/2024).

cambouropoulos˙lbdm˙2011 Emilios Cambouropoulos. "The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing". In: *International Computer Music Association* (2011). ISSN: 2223-3881.

lucas˙ising˙2014 Andrew Lucas. "Ising formulations of many NP problems". English. In: *Frontiers in Physics* 2 (Feb. 2014). Publisher: Frontiers. ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005. URL: https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2014.00005/full (visited on 10/14/2024).

huang˙towards˙2012 Jiun-Long Huang, Shih-Chuan Chiu, and Man-Kwan Shan. "Towards an automatic music arrangement framework using score reduction". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 8.1 (Feb. 2012), 8:1–8:23. ISSN: 1551-6857. DOI: 10.1145/2071396.2071404. URL: https://dl.acm.org/doi/10.1145/2071396.2071404 (visited on 12/05/2024).

li˙entropy˙2019 You Li et al. "Automatic Piano Reduction of Orchestral Music Based on Musical Entropy". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. Mar. 2019, pp. 1–5. DOI: 10.1109/CISS.2019.8693036. URL: https://ieeexplore.ieee.org/document/8693036 (visited on 12/27/2024).

owsinski˙mixing˙2017 Bobby Owsinski. *The Mixing Engineer's Handbook*. Bobby Owsinski Media Group, 2017. ISBN: 978-0-9985033-4-9.

A Code overview

B Examples

Start with an undirected graph $G = (V, E)$ and a set of n colours, which represent the parts for which we are arranging, with each being an independent set. Denote $x_{v,i}$ to be a binary variable such that

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

which requires nV variables. The energy is

$$H = H_A + H_B + H_C + H_D\,.$$

Each vertex is coloured exactly once:

$$H_A = A \sum_{v \in V} \left(1 - \sum_{i=1}^{n} x_{v,i} \right)^2$$

Vertices of the same colour are not connected by an edge:

$$H_B = B \sum_{(u,v) \in E} \sum_{i=1}^{n} x_{u,i} x_{v,i}$$

Maximise the weighting of selected vertices:

$$H_C = -C \sum_{v \in V} \sum_{i=1}^{n} W_v x_{v,i}$$

Maximise the weighting of included edges:

$$H_D = -D \sum_{(u,v) \in E} W_{uv} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{u,i} x_{v,j}$$

It can be seen that for $n = 1$ this reduces to the MIS problem. For a score with p parts, it will be impossible to colour the graph exactly with $n < p$; the parameter A should be small enough to allow for some vertices to remain uncoloured. The lowest energy solutions will return coloured independent subsets of G that each represents a monophonic part of the final arrangement.

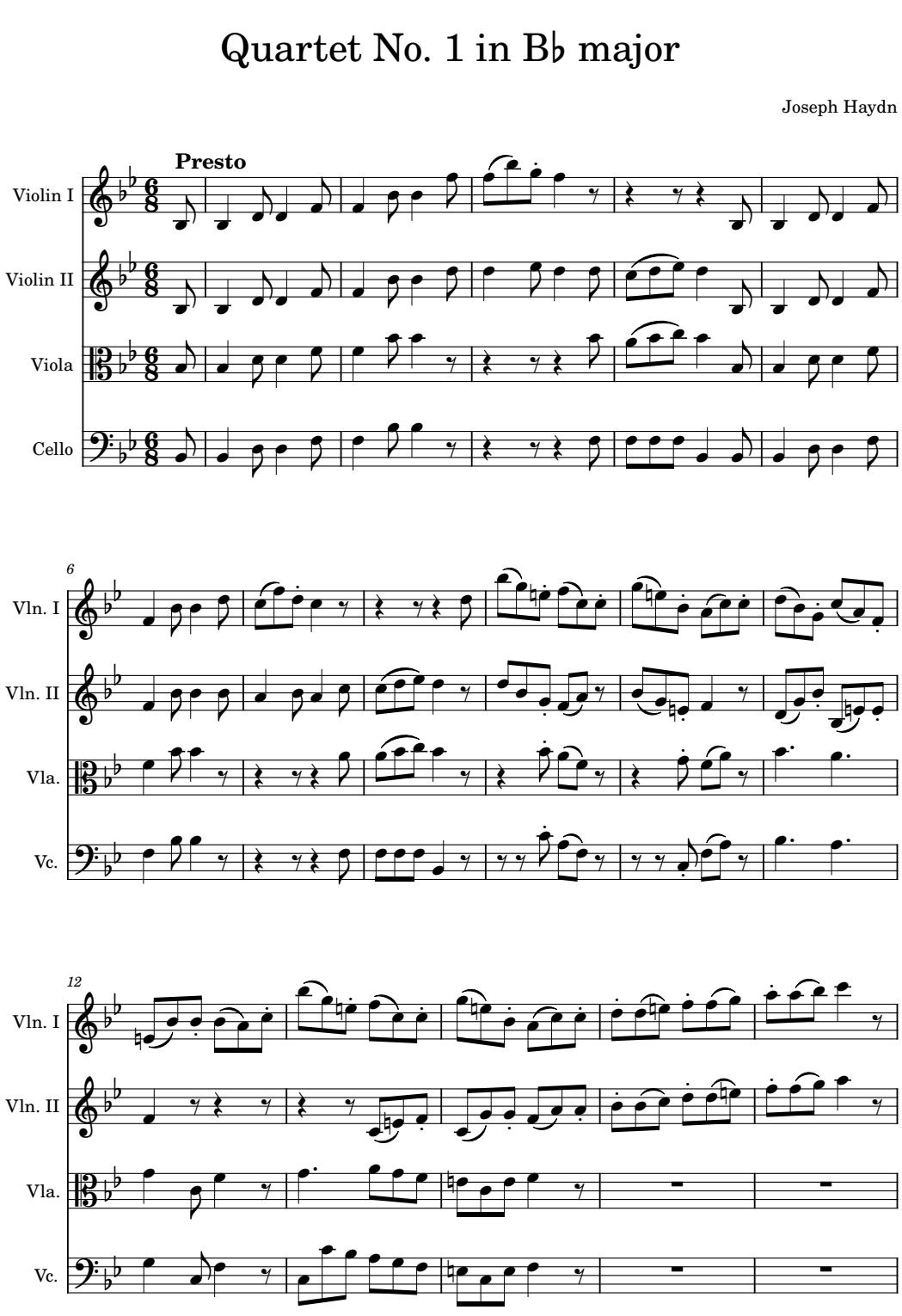

Figure 12: Quartet No. 1 in B-flat major, Op. 1, by Joseph Haydn, bars 1–16.

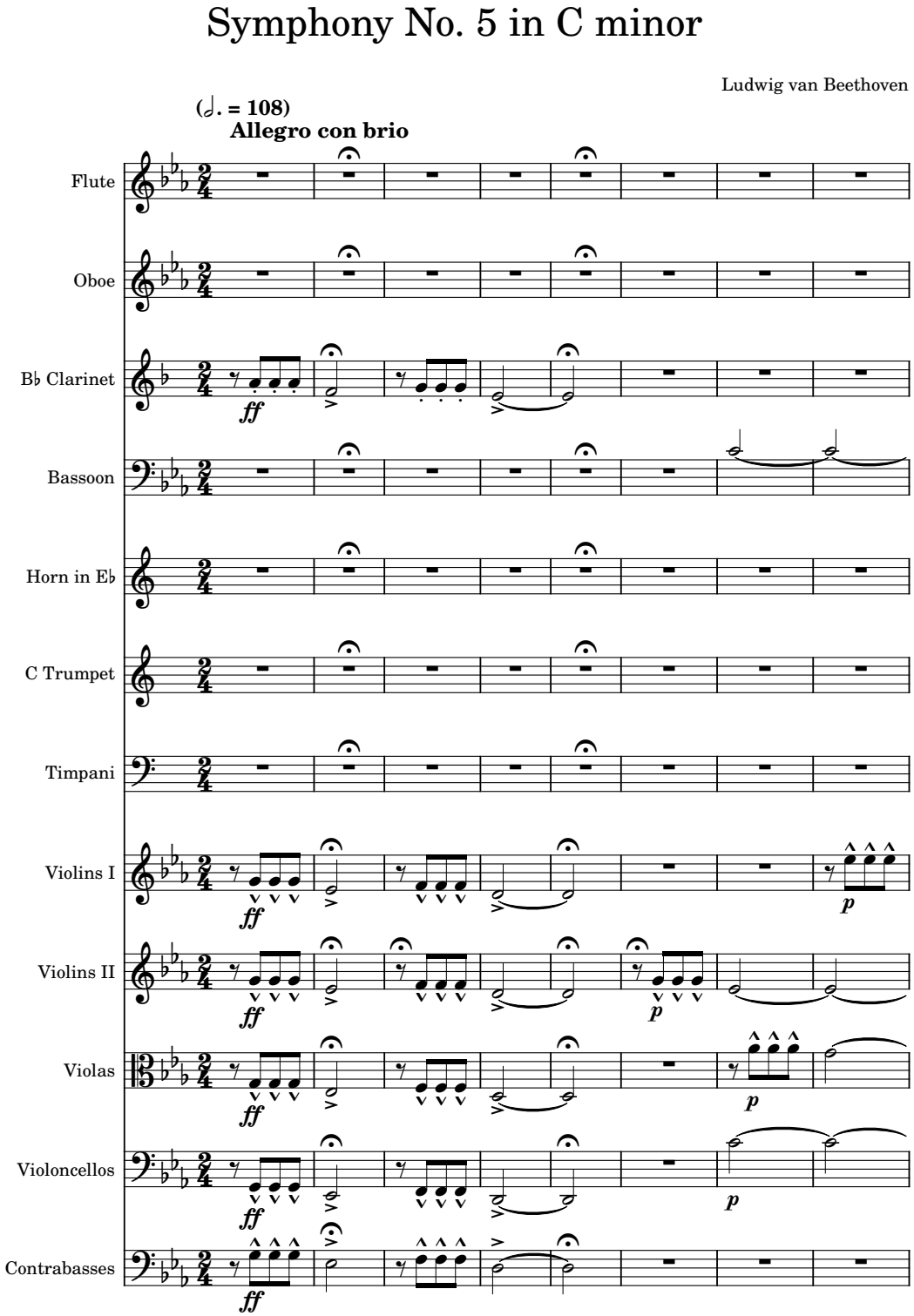

Figure 13: Symphony No. 5 in C minor, Op. 67, by Ludwig van Beethoven, bars 1–8.

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.