Towards the automatic arrangement of music via quantum annealing

L. Kirby
Durham University
Submitted: 3rd January 2025

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

I. INTRODUCTION

Music is often seen as a very human endeavour.

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, a challenging and often time-consuming process, requiring a great deal of skill and experience. 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. 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.

With the advent of quantum computers, there become growth in a new field of quantum computer music.

The combination of quantum computing and music is a relatively new field, but one that has shown promise. Quantum computers have the potential to solve complex optimisation problems that are intractable for classical computers, and music arrangement is one such problem.

Quantum computing comes in two flavours: gate-based and annealing. Gate-based quantum computers, such as those developed by IBM and Google, use quantum gates to manipulate qubits and perform calculations. Quantum annealers, such as those developed by D-Wave, use quantum fluctuations to find the global minimum of a given function. Each are suited to solving different classes of problems, with gate-based computers being more versatile and annealers being more efficient for certain optimisation problems.

Computational complexity "One of the roles of computational complexity theory is to determine the practical limits on what computers can and cannot do". Quantum annealers are particularly well-suited to solving a class of problems known as NP-hard. A well-known example of an NP-hard problem is the travelling salesman problem, where the goal is to find the shortest route that visits a set of cities exactly once. This problem is difficult to solve because the number of possible routes grows exponentially with the number of cities. Quantum annealers have been shown to be effective at solving NP-hard problems, and have been used to find solutions to a variety of optimisation problems, such as protein folding [CITE], financial portfolio optimisation [CITE], and traffic flow optimisation [CITE].

This paper will focus on a subset of arrangement known as *reduction*—taking a score written for a large number of parts and reducing it to a smaller number of parts, whilst still retaining the musicality and structure of the original. This is a common task in music production, where a piece written for a full orchestra may need to be arranged for a smaller ensemble, such as a string quartet. This forgoes the need to generate new music, as all notes in the arrangement are taken from the original score, and is more readily available to analyse as an optimisation problem. Quantum music generation is currently a developing field [1], but beyond the scope of this paper.

II. THEORY

Annealing, in metallurgical terms, is the process of heating and cooling a material to alter its physical properties. The cooling stage is typically slow, allowing particles to arrange themselves into regular lattice structures, making the material softer and less brittle [2]. In the context of quantum computing, quantum annealing is the process of slowly changing a quantum system from one Hamiltonian to another. Much like its metallurgical counterpart, this computational technique allows the system to settle into a more useful final state.

The typical time evolution of a quantum annealer looks like

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

where H_0 is the initial ground state, which is allowed to evolve to a second state H_p across time T. If this is done adiabatically, with a sufficiently large T, the system will remain in the ground state of $H(t)$, and therefore end in the ground state of H_p at time T. This is the principle of quantum annealing.

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 we prepare H_p such that its ground state encodes the solution to the optimisation problem, then as long as H_0 and H_p do not commute, the solution will be given at the end of the annealing process.

Problem Hamiltonians take the form of an Ising spin glass, a random arrangement of "magnetic dipole moments of atomic spins that can be in one of two states", typically spin up or down. A spin glass with N spins s_i takes the form

$$H_p = -\sum_{i<j} J_{ij} s_i s_j - \sum_{i=1}^{N} h_i s_i \tag{2}$$

where J_{ij} are the coupling strengths, and h_i are the field strengths. In the quantum case, the discrete variables are called *qubits* and the spins s_i are replaced with Pauli matrices σ_i^z. The ground state is prepared with the Hamiltonian

$$H_0 = -h_0 \sum_{i=1}^{N} \sigma_i^x \tag{3}$$

which is an "equal superposition of all possible states in the eigenbasis of H_p".

III. EMBEDDING

The mapping of a logical problem to a spin glass state is called *embedding*. Problems are formulated in an analogous way to a spin glass via a QUBO (quadratic unconstrained binary optimisation). A QUBO takes the form of a function $f(x)$ to be minimised, and is represented by

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j \tag{4}$$

where x is a vector of binary variables, and Q is an $N \times N$ upper-diagonal matrix of real weights. As its name suggests, a QUBO can only have terms up to the second power, restricting coupling to pairwise interactions [WHY?]. This form can easily for transformed to an Ising model using

$$s_i = x_i - \frac{1}{2} \tag{5}$$

Once a QUBO has been expressed, it can be sent to a quantum processing unit (QPU) to be solved. The graph architecture of these units allows the mapping of problem QUBOs to physical qubits, with linear terms as nodes and quadratic terms as couplers. However, the topology of these graphs often don't allow an exact one-to-one mapping; the solution to this is the introduction of chains. A chain of physical qubits can act like a single logical variable, with the chain strength determining how strongly the chain is coupled. Chain strength is an important parameter that can affect the quality of solutions [DISCUSS MORE].

The qubits and their couplers define what can be seen as an energy landscape. Different combinations of qubit spins will have different energies, with the lowest energy state being the solution to the problem. A solver (or "sampler") attempts to characterise the shape of the energy landscape using a random range of input, and returns a number of samples with the lowest energy states. Note that the "true" ground state cannot be guaranteed; it could be part of the returned sampleset, or may not even be found at all. It is the parameters of the annealing process that can reduce local minima and increase the likelihood of finding the ground state.

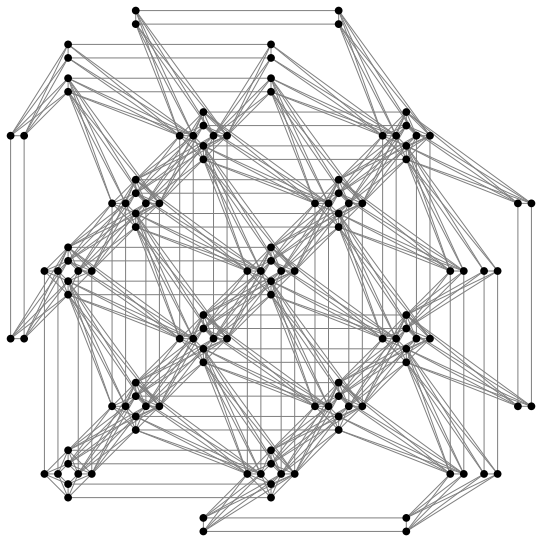


Figure 1: A graph of 144 qubits in a D-Wave QPU, using the Pegasus topology. The qubits are represented by the nodes, and the couplers by the edges.

The full embedding process is often handled entirely by the QPU, done "on the fly" each time a problem is submitted. Fixed embeddings can be specified, but these require a priori knowledge of the specific QPU architecture. The D-Wave QPU used in this study is the Advantage system, which has a Pegasus architecture. This is a highly connected graph, with 15 qubits per unit cell, and 16 unit cells. The Pegasus graph is a two-dimensional lattice, with each qubit connected to 15 others. This allows for a high degree of connectivity, but also introduces the possibility of long chains, which can degrade the quality of solutions.

IV. PROBLEM FORMULATION

Previous approaches to the automatic arrangement of music have been based on the use of genetic algorithms and neural networks. 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 approach to the automatic arrangement of music based on quantum annealing.

It has been shown that the automatic arrangement of music can be formulated as a combinatorial optimisation problem [3]. Given a set of musical phrases, the goal is to find an optimal arrangement of these phrases such that the overall quality of the arrangement is maximised, whilst considering the compatibility of the phrases. This problem can be formulated as a maximum independent set (MIS) problem, which is a well-known optimisation problem. The MIS problem can then be solved using quantum annealing, which already has well-known QUBO formulations [4].

Given a graph G with nodes x_i and edges E, the aim is to find the largest set of nodes such that no nodes within the set are connected by an edge. Formally, this can be written as

The maximum independent set (MIS) QUBO takes the

form

$$f(x) = A \sum_{ij \in E} x_i x_j - B \sum_i x_i \,, \tag{6}$$

where we take $x_i = 1$ as the inclusion of that node in the solution set. The first term penalises any edges in the solution by increasing the energy by A, whilst the second term is the objective term, that is, the quantity we are trying the minimise.

This can further be expanded to include weights on the nodes, which can be used to represent the quality of the phrases. The MIS QUBO then becomes

$$f(x) = A \sum_{ij \in E} x_i x_j - B \sum_i W_i x_i \,, \tag{7}$$

where W is a vector of real weights.

V. METHODS

There are many formats that allow the digital manipulation of music, each with their own benefits—this study primarily uses MusicXML [5]. This format focuses on the representation of standard sheet music, allowing for musical notes, rests, and other symbols, as well as the structure of the music, such as the arrangement of notes into measures (bars) and parts. It is widely supported by music notation software, allowing translation of music between graphic, textual, and audio formats.

Since interaction with the D-Wave QPUs is handled in Python, the surrounding manipulation of music is also written in this way. The Python module `music21` provides extensive resources for manipulating, translating, and creating music, allowing scores to be broken down and reconstructed as fit.

To begin processing, a MusicXML file of the original score is first interpreted by the program. Embedded metadata within the score allows it to be broken down into its constituent instrument parts, measures, 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. [CAN ELABORATE HERE]

To construct the graph, each part needs to be split into a sequence of musical phrases. The approach taken here is the local boundary detection model (LBDM) [Cambouropoulos 2001] due to its easy implementation and good performance. The boundaries segmenting a musical line into phrases are identified using the degree of change between successive notes; larger differences between notes would show an increased likelihood of a boundary. Boundary strengths are calculated over two parameters, pitch and inter-onset interval (IOI), which is the difference between the starts of two notes. This is chosen over note duration as it is often more noticeable to the listener [CITE]. The strength at note i with parameter x_i is given by

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

where $r_{i,i+1}$ is the degree of change of a parameter between notes i and $i+1$ and 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 signal the start of a new musical phrase. Each parameter's strength is normalised to the range $[0, 1]$ according to the set of all strengths S via

$$S_i' = \frac{S_i - \min(S)}{\max(S) - \min(S)} \tag{10}$$

Finally, to find the total boundary strength, the strengths of each parameter are summed with a weighting, using weights derived by trial-and-error (0.3 for pitch and 0.6 for IOI) [CITE]. If a note's total boundary strength surpasses a specified threshold, it is considered a boundary. Boundaries are always taken at the beginning and end of a part.

Once a list of boundaries is identified, the phrases are extracted by taking all the elements between subsequent boundaries. Each phrase is labelled according to the part it belongs to and its index within that part, allowing the phrases to be easily referenced and reconstructed into a new score once the optimisation is complete.

In order to construct the graph, each phrase is checked against every other phrase for an overlap in time. Phrases are then represented by nodes, with edges connecting phrases that overlap. The phrase nodes can be assigned a weighting based on their entropy. The entropy of a phrase can be seen as how much musical information it contains, with a higher entropy indicating a more complex phrase. Maximisation of phrase entropy can be used to create richer arrangements. For a discrete random variable X with a probability distribution $P(X)$, the Shannon entropy is defined as

$$H(X) := -\sum_i P(x_i) \log_2 P(x_i) \tag{11}$$

where each x_i is a possible value of X. In this context, the random variable is a musical note, considering its possible values both in terms of pitch and rhythm.

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

$$P(x_i) = \frac{n_i}{N} \tag{12}$$

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" in the final arrangement.

Rhythm entropy is calculated in a similar way, but instead of considering the duration of notes, the inter-onset interval (IOI) is used instead. This is also calculated using Eq. 12, but instead considering x_i as possible IOI values. The total entropy of a phrase is then the sum of the pitch and rhythm entropies.

Once the graph is formed, the weighted MIS QUBO can be constructed via Eq. [EQUATION] and sent to the D-Wave QPU to be solved. The solutions returned are sets of phrases that are compatible with each other and form the final arrangement. These are then reconstructed into a new object which can be translated into sheet music or audio.

Table I: Summary of frational changes to best-fit parameters due to electron energy density and electrostatic corrections.

Correction	$\lvert\Delta A\rvert/A$	$\lvert\Delta B\rvert/B$	$\lvert\Delta q\rvert/q$
$\varepsilon_{\text{elec}}$	0.026	0.0031	0.0442
p_c	0.009	0.0114	0.0005

VI. RESULTS

The method was tested on an excerpt of String Quartet No. 10 in E-flat major, Op. 74, by Ludwig van Beethoven, nicknamed the "Harp" quartet. This was chosen due to its relatively simple structure and smaller instrumentation, making it easier to identify and arrange the musical phrases. Fig. 6a shows the first 13 measures of the original score.

Phrases identified using the LBDM model are shown in Fig. 6b, alongside their calculated entropies according to Eq. 11. A graph representation of the phrases can be seen in Fig. 3, with the nodes representing the phrases and the edges connecting overlapping phrases. Many of the nodes form cliques of three or four vertices, that is, each node of the subset is connected to every other node. This can be expected, as often musical phrases across instruments will occur simultaneously, especially in small ensemble music such as this. Notably, there are also two disjoint cliques, which can be seen as distinct musical ideas that are separtated by rests from the other phrases (for example, the first two measures of the piece). A non-negligibile number of phrases are calculated to have zero entropy—these are phrases identified as only having one note, and thus no variety in pitch or rhythm.

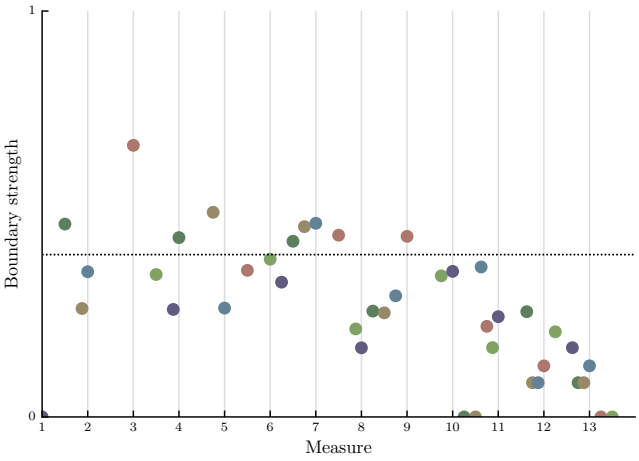


Figure 2: Boundary strengths for each note of the viola part, calculated using Eq. 8. The dashed line shows the threshold value used (0.4). Notes above the threshold value are considered as boundaries for phrases.

After submitting a QUBO calculated using Eq. 7, the D-Wave QPU returned a sampleset of solutions, which can be seen in Fig. 4. Many solutions had the same energy but used slightly different combinations of phrases to make the maximal independent set; this degeneracy is likely due to the high number of cliques, which only allows one node of the subset to be picked, and phrases with zero entropy, which do not change the final energy when interchanged. One of the lowest energy solutions can be seen in Fig. 5, with the nodes selected by the QPU highlighted. In this solution, none of the nodes are connected by edges, providing a valid arrangement. The musical score of this solution is shown in Fig. 6d.

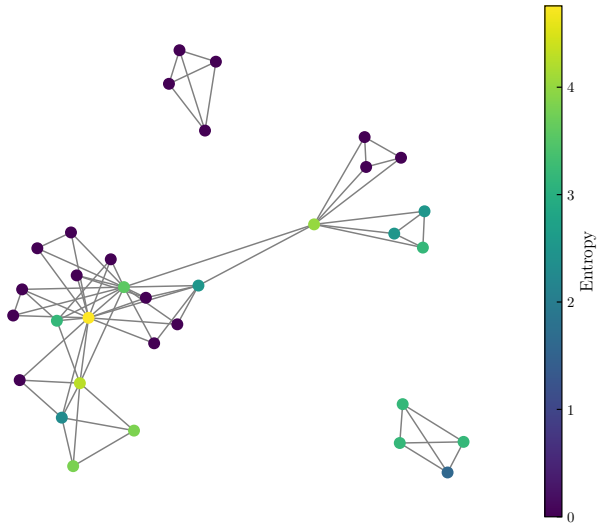


Figure 3: The representation of Fig. 6b as a graph. The nodes are the identified musical phrases, with edges connecting them if they overlap. The entropy of each phrase is represented by the colour of the node.

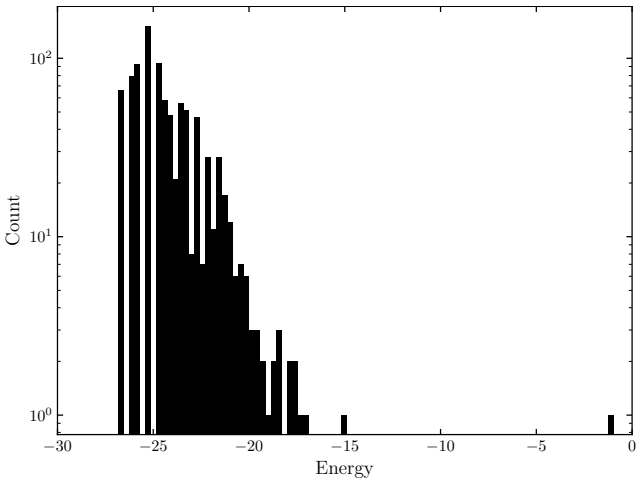


Figure 4: A histogram of the solutions returned by the QPU, over 1000 total reads. Solutions with energies above zero are omitted.

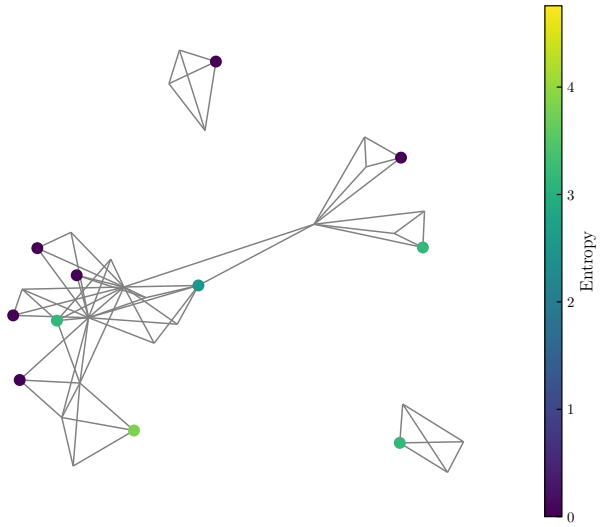
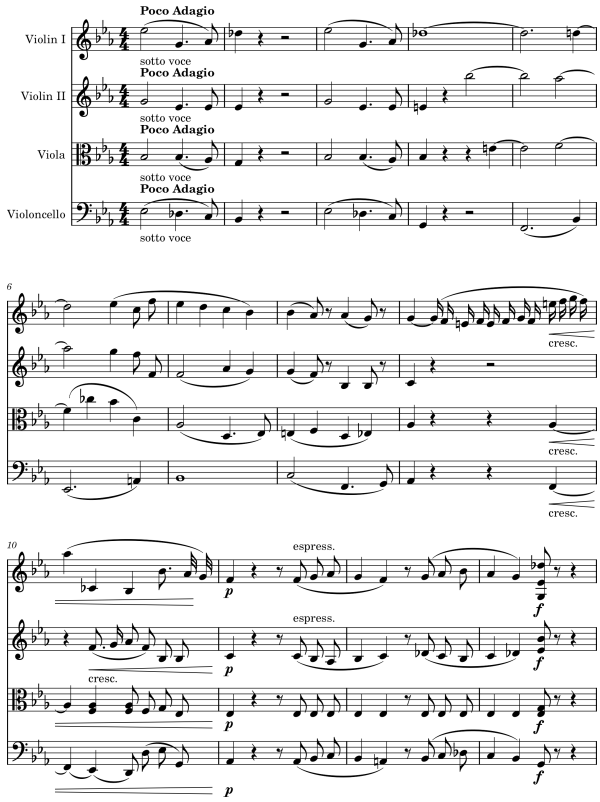
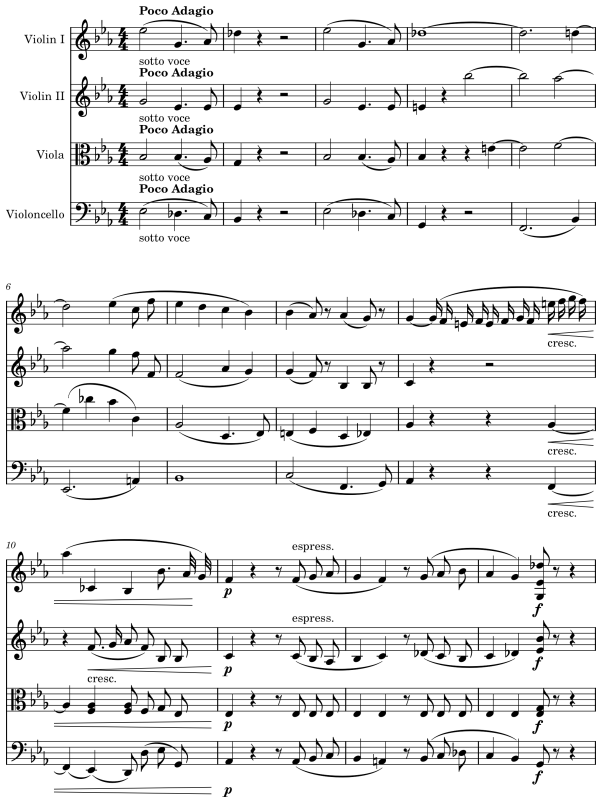


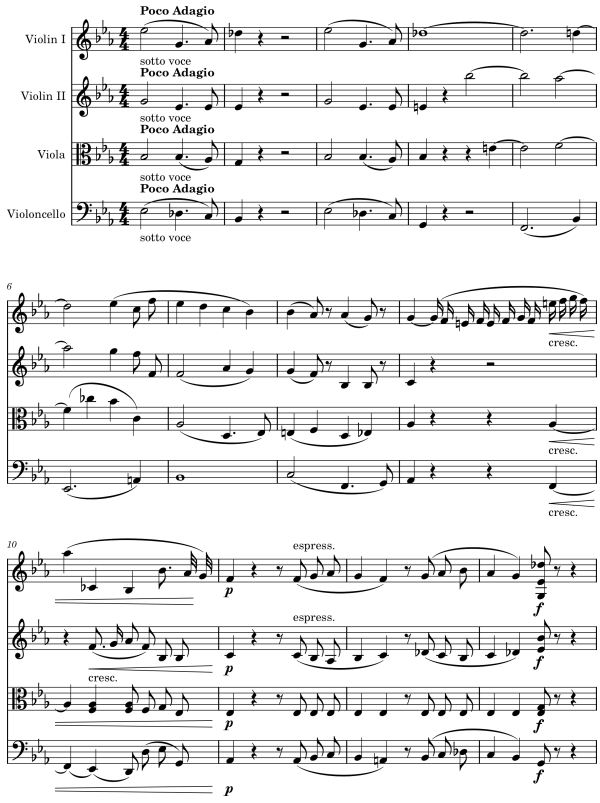
Figure 5: The graph of Fig. 3 but with the nodes selected by the QPU highlighted. In this solution, none of the nodes are connected by edges, providing a feasible arrangement, which can be seen in Fig. 6d.



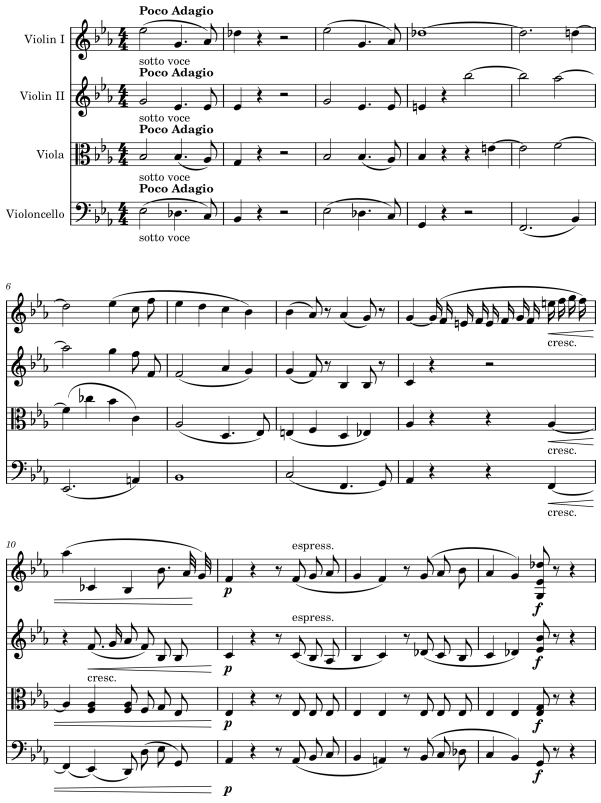
(a) Original excerpt.



(b) Musical phrases are identified and labelled with their entropy.



(c) Selected phrases from solving the MIS problem.



(d) Final arrangement.

Figure 6: The beginning of String Quartet No. 15 in A minor, Op. 132, by Ludwig van Beethoven.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

VII. CONCLUSIONS

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, using considerable time and resources. Due to the required constraints, the QPU will (mostly) always provide a feasible solution, without any knowledge of previous arrangements.

Another benefit is the solve time. For 1000 reads, the D-Wave QPU access time averages at 163 ms

Future work would include expansion to polyphonic music, allowing more than one phrase to be played simultaneously. This could take the form of arranging for a polyphonic instrument such as the piano, or for multiple monophonic instruments. The MIS formulation would need to be altered to a colouring one, with the different colours representing allocations of phrases to each ”track”.

REFERENCES

[1] Eduardo Reck Miranda, editor. Quantum Computer Music: Foundations, Methods and Advanced Concepts. Springer International Publishing, 2022.

[2] Oxford English Dictionary. annealing, n., sense 3, December 2024.

[3] William S. Moses and Erik D. Demaine. Computational Complexity of Arranging Music, July 2016. arXiv:1607.04220.

[4] Andrew Lucas. Ising formulations of many NP problems. Frontiers in Physics, 2, February 2014. Publisher: Frontiers.

[5] MusicXML for Exchanging Digital Sheet Music.

[6] Jiun-Long Huang, Shih-Chuan Chiu, and Man-Kwan Shan. Towards an automatic music arrangement framework using score reduction. ACM Trans. Multimedia Comput. Commun. Appl., 8(1):8:1–8:23, February 2012.

[7] Eita Nakamura and Kazuyoshi Yoshii. Statistical piano reduction controlling performance difficulty. APSIPA Transactions on Signal and Information Processing, 7:e13, January 2018.

[8] Hao-Wen Dong, Chris Donahue, Taylor Berg-Kirkpatrick, and Julian McAuley. Towards Automatic Instrumentation by Learning to Separate Parts in Symbolic Multitrack Music, October 2021. arXiv:2107.05916 [cs].

[9] Moyu Terao, Eita Nakamura, and Kazuyoshi Yoshii. Neural Band-to-Piano Score Arrangement with Stepless Difficulty Control. In ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5, June 2023. ISSN: 2379-190X.

[10] Yuki Hoshi, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga. Versatile Automatic Piano Reduction Generation System by Deep Learning. In 2022 2nd International Conference on Advanced Research in Computing (ICARC), pages 66–71, February 2022.

SCIENTIFIC SUMMARY FOR A GENERAL AUDIENCE

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.