A Four-Part Harmony Composition Solver Based on Bayesian Framework

Xiaolong Ma (mxl20@mails.tsinghua.edu.cn)[1]

[1]**Tsinghua University**, Beijing, China

ABSTRACT

As a foundational lesson for beginner music composition students, Four-part Harmony Composition involves composing the other three parts to complement a given Soprano melody. However, the task is challenging due to basic rules that govern such compositions.The automation of Four-Part Harmony Composition problems has been a topic of discussion for some time, with various attempts made to achieve this objective. Unfortunately, the effectiveness of these approaches remains debatable, with little flexibility applied in previous frames.

We have extended the rule-based Bayesian method to recognize chord termination and to include sixth, sixth-four, and ninth chords and their inversions in our compositions. Our Four-part Harmony Composition solver can handle typical 8-bar compositions with ease, and users can customize their fixed chord settings for added flexibility and user-friendliness. We believe our work will be of great help to both music composers and learners, in the hope that composers can acquire inspirations by the outcome of our algorithm.

1. INTRODUCTION

The most ancient form of music is monophonic melody, which features only a solo singer or a single instrument with no accompaniment. Later, harmony was developed, where a group of notes sound together, creating more colorful music. Harmony has several common types, such as triad, sixth chord, seventh chord, and ninth chord, with their inversions. A triad is a three-note chord built on thirds. For example, in C Major, the notes do(C), mi(E), and so(G) form the *tonic chord*, while in A Minor, the notes la(A), do(C), and mi(E) form the tonic chord. Seventh and ninth chords consist of four and five notes, respectively. Chords can be inverted to provide variations in music. The first inversion of a triad is called a sixth chord, with the original middle note at the bottom. The second inversion of a triad is called a sixth-four chord, with the original top note at the bottom. Chords are more colorful than monophonic melody, and each type of chord and inversion convey different emotions. The progression of chords illustrates how emotion changes, so it is crucial for composers to choose appropriate chords and progression that effectively communicate their emotions to the listener.

Figure 1. A melody in D Major.

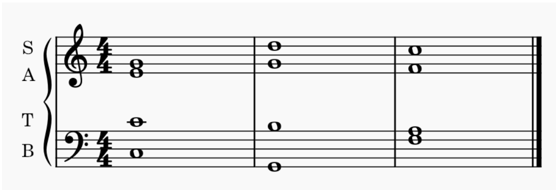

Figure 2. A close tonic, an open dominant chord, an open subdominant chord in C Major, from left to right respectively.

There are several fundamental lessons for music composition students, among which Four-part Harmony Composition is usually the first composition lesson, training students with Basic music theory knowledge. We can easily tell from the name that there are four parts in a Four-part Harmony, that is Soprano, Alto, Tenor and Bass. The names of the four parts come from chorus, where Soprano, Alto, Tenor and Bass refer to women singers with high singing voice, women singers with low singing voice, men singers with high singing voice, and men singers with low singing voice respectively. In a Four-part Harmony Composition problem, usually the Soprano, that is the melody, is given, and the composer needs to compose the other three parts, ensuring the music is pleasing and consistent. Usually the task is not as easy as it sounds to be, because there are quite a few rules that govern such compositions. For instance, the intervals between adjacent parts (Soprano and Alto, Alto and Tenor) should be no more than one octave. Otherwise, the chord sounds empty to some extend. Apart from that, Alto and Tenor should not leap greater than perfect fourth, because this would guarantee the inherence of the music. Some of the rules are easy to ignore, while some may be difficult to obey. Therefore, it usually takes quite a while to solve a single 8-bar long Four-part Harmony Composition problem perfectly for a beginner.

There are several prior works on automatic composition of Four-part Harmony utilizing different methods. Some are based on traditional machine learning techniques, and some apply deep learning.

In our work, we mainly focus on the typical Four-part Harmony Composition, where Soprano is already given and the other three parts remain to be composed. And for convenience, we limit the problem length to 4 or 8 bars, so there are usually two cadences, one at the end and the other at the end of the fourth bar. We have complemented a rules-based bayesian method to solve Four-part Harmony Composition problems. Besides, users can customize their own fixed chord settings, and our system can generate an appropriate solution according to the users' preference. Compared to prior methods, our method shows more robustness and flexibility in 8-bar long Four-part Harmony Composition problems, and includes more types of chords. We successfully include triads, seventh chords, ninth chords and their inversions in our composition, and composers can get a solution with no violation of all the constraints.

2. RELATED WORK

Generally, our work belongs to automatic music generation, and an increasing number of researchers have dived into this field due to the flourish of deep neural network. We will introduce related work in the following two subsections, *Four-part Harmony Composition* and *Music Generation*. In the first subsection, we will review on similar work on Four-part Harmony Composition, while in the second , we will briefly look at some latest work in the music general field.

2.1 Four-part Harmony Composition

Various attempts have been made to solve the Four-part composition problems automatically. Somnuk et al. proposed a Rule-Based system [1]. Different from their work, we mainly apply a bayesian framework to include the rules while they simply impose penalties if a rule is violated. Various machine learning algorithms have been attempted. Marco el al. used a Multiple Worlds Model for such tasks [2]. Donnelly et al. studied the application of genetic algorithm in four-part harmony [3], and similar for Yi el al. [4]. Some surveys in musical harmonization with constraints have comprehensive insight into the problem we are discussing, like [5] [6]. Indeed, Four-part Harmony Composition can be solved in these machine learning algorithms but we believe a bayesian model will guarantee complying with the rules more easily.

Neural Network has been applied to this generation task [7]. Obviously it is a sequence generation task, so a recursive neural network(RNN) could be a good choice. One tough problem is that there isn't a large enough dataset. Unlike other generation task, Four-part Harmony Composition has strict rules that govern such generation, so a generative neural network can make a mistake if the dataset is not large enough.

2.2 Music Generation

As deep neural network is flourishing these years, there are a lot of impressive work on music generation. GAN [8], VAE [9], Diffusion Network [10] are some representative generative models, which have achieved astonishing results. The boom is even more intense when GPT series [11] in natural language process field (NLP) and DALLE2 [12] in computer vision (CV) have come out, and generative models have reached an accelerated stage of development. Some music generative models have been published such as the model noise2music [13] which can turn a bunch of random noise into pleasing music step by step through a diffusion model.

Unfortunately, music generation does not yet have a large enough model that can fit in every task, and audio domain has a large room for development compared to literal domain NLP and visual domain CV. We hope the universal grand model of music generation will come soon.

Due to lack of dataset, and strict rules in Four-part Harmony, our work is restricted to traditional bayesian methods. It's more than adequate for us to have achieved nearly perfect performance in the generation task, and it would be more flexible and easy to adjust to apply a bayesian framework here, which means users can adjust the music style according to their own preference, but we will still explore some generative network in our future work.

3. METHOD

3.1 Bayesian Method

Bayesian Method [14] is one of the most typical methods in sequence generation, where we focus on the probability that two or more particular chords are adjacent to each other. In our work, the input of the system is a series of notes, i.e. the Soprano part, and what we hope to get is a series of chords, i.e. a finished Four-part Harmony Composition. We can view input and output as sequences of the same length, as in our case we will give each note a single chord. Certainly it is quite common for one note to correspond to multiple chords, but that is easy to complement if we simply treat one note as two. Let's assume the input sequence is $\{A_n\}_L$, where L represents the length of the input, and assume the output sequence to be $\{B_n\}_L$. Then we can calculate the possibility of producing a sequence $\{B_n\}_L$, which is formula(1)

$$\begin{aligned} p(B_1B_2\cdots B_L) =& p(B_1)\times p(B_1|B_2)\times \\ & p(B_1B_2|B_3)\times \\ & \cdots\times \\ & p(B_1B_2\cdots B_{L-1}|B_L) \end{aligned} \tag{1}$$

Then the objective turns into maximizing $p(B_1B_2\cdots B_L)$. It has to be emphasized that all the possibility expressions above, i.e. $p(\ldots)$ are under the condition that $\{A_n\}_L$ is given, so the possibility expressions of series $\{B_n\}_L$ are dependent on $\{A_n\}_L$ though the latter is not explicitly contained in the expression formula(1). We can simplify the formula to get the following form(2), considering that most chord progression rules are about adjacent chords.

$$\begin{aligned} p(B_1B_2\cdots B_L) =& p(B_1)\times p(B_1|B_2)\times \\ & p(B_2|B_3)\times \\ & \cdots\times \\ & p(B_{L-1}|B_L) \end{aligned} \tag{2}$$

This is a typical binary model in Bayesian Method, which is a usual way to deal with sequence generation task where the task can be viewed as a markov chain. We can easily derive the ternary model in the same way. See equation(3)

$$
\begin{aligned}
p(B_1 B_2 \cdots B_L) = & p(B_1) \times p(B_1|B_2) \times \\
& p(B_1 B_2|B_3) \times \\
& p(B_2 B_3|B_4) \times \\
& \cdots \times \\
& p(B_{L-2} B_{L-1}|B_L)
\end{aligned}
\tag{3}
$$

In our method, we manually define each transition probability from one chord to another. Indeed that's a cumbersome and massive project, and these efforts to quantify possibility can easily lead to subjectivity. As a result we did spend a long time adjusting all the manually set possibility according to the final output. We will introduce the settings more specifically in the following subsections.

3.2 Chord Arrangement Rules

When we compose with chords, basically there are two problems ahead of us, that is what type of chords to choose and how to arrange them. Given a fixed note in the Soprano part, composers need to figure out what function of chord can be used here, according to the position of the note in the melody and the conveyed emotion. For instance, when a so(G) appears at the end of a melody of C major, a tonic chord, i.e. do(C), mi(E), so(G) is preferred because a tonic chord can indicate an end. Though a dominant chord, i.e. so(G), xi(B), re(D) also contains so(G), but audience will always expect a tonic chord after a dominant chord, so more often than not a dominant chord can't be used to imply the end of the music.

After we have determined the function of the chord, another problem to consider about is the arrangement of the chord. A triad can be arranged in two ways: open and close. A close arrangement means the three triad notes are adjacent corresponding triad notes on a piano while an open arrangement has nonadjacent notes, as we can see in figure 2. Other chords like seventh chords and ninth chords are far more complicated as they have more variations in arrangement, and different inversions.

When solving a Four-part Harmony Composition problem, we take the following chords into consideration:

- tonic triad(T)
- dominant triad(D)
- subdominant triad(S)
- supertonic triad($\mathrm{TS_{VI}}$)
- sixth chord($\mathrm{T_6}$, $\mathrm{D_6}$, $\mathrm{S_6}$, $\mathrm{S_{II_6}}$, $\mathrm{TS_{VI_6}}$)
- sixth-four chord($\mathrm{T_6^4}$, $\mathrm{D_6^4}$, $\mathrm{S_6^4}$)
- cadential six-four chord($\mathrm{K_6^4}$)
- dominant seventh chord($\mathrm{D_7}$)
- subdominant seventh chord($\mathrm{S_7}$)
- supertonic seventh chord($\mathrm{S_{II_7}}$)
- leading seventh chord($\mathrm{D_{VII_7}}$)
- chord of the second($\mathrm{D_2}$,$\mathrm{D_{VII_2}}$, $\mathrm{S_{II_2}}$)
- four-three chord($\mathrm{D_3^4}$, $\mathrm{D_{VII_3^4}}$, $\mathrm{S_{II_3^4}}$)
- six-five chord($\mathrm{D_5^6}$, $\mathrm{D_{VII_5^6}}$, $\mathrm{S_{II_5^6}}$)
- dominant ninth chord($\mathrm{D_9}$)

3.3 Chord Progression Rules

Chord Progression means the variations of selected chords when music proceeds. A good progression conveys appropriate emotions to audience, as well as sounds beautifully and consistently. Indeed music appreciation is subject to everyone, to a large degree, which is probably a consensus, but there is alway good music that can please the vast majority. Musicians have summarized some basic rules in composition which usually lead to a pleasant and intoxicating sound to whoever the listener is, no matter what type of music they like and whether they have knowledge in musical theory. These rules describe how chords should be arranged and progress. In other words, some chord progression is preferred while some should be avoided even forbidden. These rules persisted throughout classical music. However, such rules are not absolute in modern composition, though, and we can see a lot of exceptions that don't even follow any explicit rules, but when we analyze the backbone of modern music ,we won't be surprised to find that such classical chord progression rules are implied from time to time, and quite a few contemporary musics are inspired by classical ones. Still, chord progression rules are compulsory knowledge for composition beginners. In the Harmony course which is taught in a music college, students are usually required to compose the other three parts to complement a given Soprano melody. Another common requirement is to compose with a given Bass sequence, which set the tone of the music piece. That is to say, composers can compose their own Soprano melody in the latter task. Both composition tasks require absolutely strict compliance with the chord progression rules, and each violation leads to a deduction of the final points. This is a basic skill of a composition student, and it's gained by a tremendous effort. Music strictly complying with these rules are, to some extent, pleasant to a vast majority, at least far beyond unpalatable.

Here in our method, a penalty of possibility is given when a violation of the rules happens, and penalties vary slightly according to the importance of the rules.

Here we list all the rules that govern our composition, and violations of them will get a possibility penalty of 0.05.

- Intervals between adjacent parts are no more than 1 octave
- No hidden unison
- Every part should stay in its own range
- Bass should not leap over a octave

- Seventh from dominant seventh must resolve step down
- Alto, Tenor should make harmonic progression (Progression, Retrogression, Repetition), unless there is a leap in Soprano
- A chord on a strong beat should not be the same as the chord on the last weak beat
- Interval between Tenor-Bass is no more than 12 diatonic degrees
- No parallel unison, fifth, octave between two voices
- End the phrase with a cadence
- No doubling of the leading

3.4 Other Rules Contributing to A Better Performance

Rules we illustrated in the last subsection are what we graphically call dogmatic rules, which are almost never allowed to violate when we complement a Four-part Harmony Composition task. Meanwhile, there are some empirical rules which are not explicitly written on a composition textbook, but still contribute to a better performance. These empirical rules are either on chord progression or on chords themselves, and sometimes they can be disobeyed. For example, it may sound split if the for parts progress in the same direction. Besides, a leading tone has a strong tendency to move for the tonic right above, and a dominant chord will seek for an adjacent tonic chord. Progressions that follow these rules will be given a little bonus.

Here we list the empirical rules which may contribute to a better performance:

- The four parts should not progress in the same direction
- Progression is more preferred rather than leap
- A leading tone tends to resolve to the tonic step above
- A chord in dominant function tends to resolve to a tonic rather than subdominant
- A subdominant chord should not appear at the very beginning

4. IMPLEMENTATION

Some tricks in our implementation are not involved in previous sections but they are crucial in our algorithm, so some supplementary details will be added in this section.

Different from binary and ternary bayesian model, we combined the two methods, which means we deal with different rules with various methods. If a rule is concerned with multiple previous chords, then we will check the rule strictly no matter how long the sequence piece is involved. Our method is generally less computationally intensive, so we can preserve more likely solutions while processing the problem with bayesian framework. To be specific, we preserve 10 most possible solution sequences when we enumerate possible chords according to a given note, and we abandon solutions with possibility lower than 0.5 directly.

We use python to implement our algorithm, and all our functions have been packed in a python package. We will continue to test and iterate on the algorithm, and we will open the interface or even open source when appropriate.

5. EXPERIMENTS

To illustrate the performance of our method, we randomly choose some composition exercises in a classical textbook of harmonics [15], with each exercise from a chapter in the textbook to testify a corresponding function we have claimed.

We type in the Soprano part as the input, and then our system will output ten possible solutions with highest possibility. We display the solution with the highest possibility below after drawing the chords into the stave.

In the first test, we simply try a basic problem to see whether the triad transition works, and the given melody does't include a leap. See the output in Figure 3. We can conclude easily that our methods meet all the criteria, no crossing, no leaping, no inappropriate intervals and it sounds smooth and beautiful if we play it on the piano. A composition student may get a relatively good mark if this answer is handed in.

Figure 3. A basic test for our algorithm

In the second test, we choose an exercise with a dominant seventh to be resolved.

Figure 4. A test to see whether our method handles dominant seventh properly.

We are delighted to see that our method handles dominant seventh and its resolution in a correct way, and other details including leaps are generated delicately, with no obvious drawbacks.

At last, we validate the function that our system will output robust solutions even if users set certain chords fixed. Thus, users can adjust music style according to their own preference. Note that not all settings are legal, and some settings may lead to no solution.

Figure 5 is the output of an exercise without user interference. This is a short melody in G major, and our algorithm perfectly deal with the leap connection between a triad and

Figure 5. An outcome without user interference.

a sixth chord. If the user prefers to begin with an original tonic chord, then our algorithm generates the solution shown as Figure 6. We can see that our method did well and the solution meets all the constraints.

Figure 6. The user set the first chord to be a tonic triad.

We choose another short melody which is more difficult, because there could be ambiguity in its tonality. See Figure 7. Here our algorithm recognizes it as a melody of F Major. But if we set the tonality to be minor, sat to fix a minor chord, then Figure 8 is given.

Figure 7. An outcome without user interference.

Our algorithm successfully recognizes the melody to be a minor one this time, and both solutions are beautiful to ear.

6. CONCLUSIONS

In this paper, we proposed a rules-based Bayesian framework method to solve the typical Four-part Harmony Composition problem. We can solve the 4/8-bar long problem with almost perfect solutions, and even after users set fixed chords according to their preference, our algorithm shows extremely strong robustness. We are confident that this work will be of great help for both music composers and learners.

7. LIMITATIONS AND FUTURE WORK

Though our method has got a good performance on the 8-bar long composition problems, it is still weak when dealing with problems with irregular bars. Besides, our methods obey the typical rules strictly, which guarantee a relatively good solution but may lack innovation somehow. An obvious future work is to extend our method to another type Four-part Harmony Composition problem, that is to compose three parts with a given Bass. This work would be a little bit more difficult because the Soprano, i.e. the

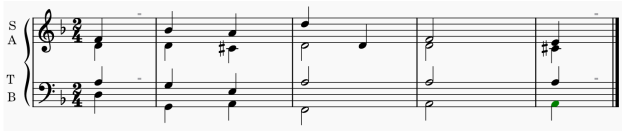

Figure 8. The user set the tune to be a minor.

melody will be generated by an algorithm. Compared to chord progression, we have less explicit rules that guarantee a melody that is pleasing to the ear.

Besides, we will open our method to public attention to gain sufficient feedbacks from different users. That way, we can adjust our possibility and rule settings more reasonable.

Also, we can include more rules to extend to more richful functionality. In the classical textbook, there are other chords and relevant rules to compose such as Secondary Dominant Chord ($\mathrm{DD}_{\mathrm{VII}_7}$, DD_9 etc.).

Furthermore, we can try other methods of calculating possibility, such as bidirectional methods. We don't have to always calculate the possibility from left to right. Instead, we can deal with the cadence at the first place. This method mimics human thought more closely, because when composers are solving the Four-part Composition problem, they do deal with the cadential chord at the first place more often than not.

Another possible future work is to include deep neural networks, especially generative networks in such problems. One possible difficulty is that we don't have large scale datasets at the moment. If we want to include Transformer-based network [16] in our problem, there is no doubt that we need a huge amount of data to train the network.

Acknowledgments

Supported by Tsinghua University Initiative Scientific Research Program (Student Academic Research Advancement Program).

8. REFERENCES

[1] S. Phon-Amnuaisuk, G. Wiggins *et al.*, "The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system," in *Proceedings of the AISB'99 symposium on musical creativity*. AISB London, 1999, pp. 28–34.

[2] M. Scirea and J. A. Brown, "Evolving four part harmony using a multiple worlds model," in *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, vol. 1. IEEE, 2015, pp. 220–227.

[3] P. Donnelly and J. Sheppard, "Evolving four-part harmony using genetic algorithms," in *Applications of Evolutionary Computation: EvoApplications 2011: EvoCOMNET, EvoFIN, EvoHOT, EvoMUSART, EvoSTIM, and EvoTRANSLOG, Torino, Italy, April 27-29, 2011, Proceedings, Part II*. Springer, 2011, pp. 273–282.

[4] L. Yi and J. Goldsmith, "Automatic generation of four-part harmony," *BMA*, vol. 268, 2007.

[5] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, "Automatic melody harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2021.

[6] F. Pachet and P. Roy, "Musical harmonization with constraints: A survey," *Constraints*, vol. 6, no. 1, pp. 7–19, 2001.

[7] T. Yamada, T. Kitahara, H. Arie, and T. Ogata, "Four-part harmonization: comparison of a bayesian network and a recurrent neural network," in *Music Technology with Swing: 13th International Symposium, CMMR 2017, Matosinhos, Portugal, September 25-28, 2017, Revised Selected Papers 13*. Springer, 2018, pp. 213–225.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[9] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[10] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[13] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, "Noise2music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.

[14] J. M. Bernardo and A. F. Smith, *Bayesian theory*. John Wiley & Sons, 2009, vol. 405.

[15] A. Schoenberg and L. Stein, *Structural functions of harmony*. WW Norton & Company, 1969, no. 478.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.