HARMONIZE THIS MELODY: AUTOMATIC FOUR-PART HARMONY GENERATION USING NEO-RIEMANNIAN VOICE-LEADING

Timothy Greer
University of Southern California
Department of Computer Science

Shrikanth Narayanan
University of Southern California
Department of Electrical Engineering

ABSTRACT

Harmonization is an important component of any comprehensive music generation system. Harmony provides context for melody and indicates the tonal framework and progression of a musical passage. One interesting element of music is that harmonic accompaniment is never strictly "wrong" or "right." In this work, we use chord choices and their progressions as proxies for harmonic movement. By using the information contained in a chord at a given moment and an input melody note, this system will output what accompanying chord could be used next in a musical passage. An additional parameter can be tweaked which determines how "conservative" the next chord choice would be, providing more possibilities for harmonization. Composers and songwriters can use this tool to brainstorm new harmonic choices, and anyone with an interest in music can find new chord changes to their favorite songs given that they have the melody notes for such a song. This work has applications in music generation and symbolic music representation.

1. INTRODUCTION

Music harmonization is a longstanding area of research in music generation [1]. Some automatic harmonization systems in the past have used Markov models [2, 3], but these models were informed by training data and only prescribed *one* harmonization output based on the inputs. Other work has used statistical inputs in conjunction with general rules of harmony to automatically create harmonizations [4]. [5] exploits part-whole hierarchies for automatic harmonization, and others have employed deep learning to harmonize using triadic approaches [6] or template methods [7].

This system uniquely blends concepts from symbolic processing and music theory, providing musicians and laypeople alike a wellspring of interesting new harmonic progressions and accompaniments for melodies. By showing a method of music generation that is informed by context-independent voice-leading principles, we open up further inquiry into studying symbolic representations of audio and automatic music generation. Our code is also available at `https://github.com/timothydgreer/4_part_harmonizer`.

Melody:	G	A	G	C	B
Harmony 1:	C6	-	-	-	Gadd2
Harmony 2:	Abmaj7	-	-	-	Emin9

Figure 1. The melody and two potential harmonic accompaniment of the beginning of "Happy Birthday." Top row: The input melody of "Happy Birthday" (G and B); middle row: The output harmony with a conservative harmonization; bottom row: the output harmony with a more complicated harmonization.

2. OUR SYSTEM

Our system algorithmically chooses sets of well-formed harmonies and harmonic transitions that can be used to accompany melody lines based on neo-Riemannian voice-leading [8], rather than depending on supervised learning or context-dependent rules of harmony, like Fux Counterpoint [9]. Given an input melody (see Figure 1), this system provides a four-part harmony to be used concomitantly with the melody (chords that could be played behind this singing), similar to [10].

Sometimes chord transitions (and, indeed, music in general) can be expected and therefore not as enjoyable [11]. [1] Inspired by the "scariness" parameter in [12], we encourage more unexpected harmonizations by including a tunable parameter we call temperature that controls how strictly the underlying harmony obeys neo-Riemannian voice-leading principles. By maximizing this parameter, harmonic accompaniments and their transitions will generally sound more iconoclastic—even dissonant—in relation to the input melody notes (see Harmony 2 in Figure 1); minimizing temperature will generally result in more conservative, static harmonizations.

[1] Oftentimes, "Happy Birthday" is sung with the exact chord choices in Harmony 1 of Figure 1.

$Chord_1$	$Chord_2$	d($chord_1$,$chord_2$)
Cmaj7	Cmaj7	0
Cmaj7	C7	1
Cmaj7	Dmaj7	6

Table 1. Chord distance based on their note interval difference. The B in Cmaj7 is one semitone away from Bb in C7, and the distance between Cmaj7 and Dmaj7 is 6: C voice-leads to C# (1 half step), E travels to D (2 half steps), G goes to F# (1 half step), and B transitions to A (2 half steps).

Tetrachord	Cmaj7	Fmin9	Dmin6	C
Note 1	C	F	D	C
Note 2	E	Ab	F	E
Note 3	G	Eb	A	G
Note 4	B	G	B	C

Table 2. Some tetrachords and their spellings. When a chord contains more than four notes, the 5th of the chord is removed (as in Fmin9). In the case of a triad, the tetrachord is doubled at the root of the chord. In order to transition to a tetrachord, it must have at least one note that matches the melody note input.

3. METHODS

3.1 Choosing Chords

3.1.1 Neo-Riemannian Voice-Leading

In this work, we use the context-independent Neo-Riemannian voice-leading system [13] to determine how similar two chords are to one another. In this system, two chords are close to each other if only a few semitonal shifts are required to transform one chord into the other [8]. We define the Neo-Riemannian distance $d(\cdot,\cdot)$ as the number of total semitonal shifts required to change one chord into another and use this metric determine how the output harmony evolves. See Table 1 for examples.

3.1.2 Chord Transitions

Inspired by [2], we use a Markov model to determine how to transition from one chord to another based on these neo-Riemannian distances.

In order to do this, we first construct a set of *tetrachords*, or four-note chords, for every natural melodic note (seven sets total—one for A, B, C,... G). See Table 2 for some tetrachords we used and their spellings. We construct stochastic probabilities of transitioning from $chord_1$ to $chord_2$, given by:

$$p(chord_2|chord_1) = \frac{(d(chord_1, chord_2)+1)^{-t}}{\sum_c (d(chord_1, chord_c)+1)^t} \quad (1)$$

where t is the parameter defined by $e^{-temperature}$ where $temperature$ is a user-inputted value between -5 and 5, c is the set of all chords containing the future melody (note the denominator of Eqn (1) is a normalizing factor), and d is the Neo-Riemannian distance.

3.2 Temperature Tuning

The temperature of the system can take on float values between -5 and 5. The value t is defined as $e^{-temperature}$. When the temperature is -5, the probability of transitioning to a chord that is dissimilar to the previous chord is very small, as large distances are inflated by a large t, bringing down the transition's probability. This corresponds to a system that conservatively chooses chords. When the temperature is 5, the probability of transitioning to a chord that is extremely different from the current chord is much higher: in this case, every possible chord transition probability is about the same, because even the large-distance transitions are brought close to 1 after being exponentiated by a small, positive t. In this way, one set of melodic notes can create vastly different harmonizations based on the temperature of the model.

4. LIMITATIONS

There are several limitations to this work. In Figure 1, for example, Harmony 2 is consonant given the melodic inputs, but it is dissonant when accounting for the melodic notes that are *not* included in the input. Here, Abmaj7 is a valid chord choice for the melodic note of G, but the melodic note A is dissonant given the Abmaj7 chord. A future iteration of this system will allow for multiple melodic inputs for each output chord a user requests, and each output will have to accommodate *all* melody notes between chord transitions, not just the first one in a melodic line.

Furthermore, only natural notes (A, B, C,...G) can be used as melodic input to our system. This was to ensure that the melodies inputted were only in the key of C major or A minor, allowing for easier construction of the set of chords that could harmonize the melodic notes. While harmonic accompaniments can contain notes or even roots that are non-natural (Emin9, for example, has an F#), [2] a user must sometimes transpose an input melody into the proper key in order to use our system. Further work will be done to expand the capabilities of this system so as to accommodate non-natural melodic note inputs.

5. CONCLUSION

Harmonization is an paramount to any comprehensive music generation system. In this work, we create possible harmonic accompaniments of input melodies. We also outfit our system with a parameter that dictates how "conservative" or "unlikely" a chord transition output will be. This tool can be found at https://github.com/timothydgreer/4_part_harmonizer and can be used by musicians to create accompaniment for their melodies or more generally anyone with a curiosity in music generation and/or symbolic processing.

[2] These harmonizations may indeed inspire new non-natural, melodic compositions as well.

6. REFERENCES

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

[2] R. Groves, “Automatic harmonization using a hidden semi-markov model,” in *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[3] M. Allan and C. K. Williams, “Harmonising chorales by probabilistic inference,” *Advances in neural information processing systems*, vol. 17, pp. 25–32, 2005.

[4] R. P. Whorley and D. Conklin, “Music generation from statistical models of harmony,” *Journal of New Music Research*, vol. 45, no. 2, pp. 160–183, 2016.

[5] F. Pachet and P. Roy, “Formulating constraint satisfaction problems on part-whole relations: The case of automatic musical harmonization,” in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI 98)*. Citeseer, 1998, pp. 1–11.

[6] 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.

[7] M. McVicar, R. Santos-Rodríguez, Y. Ni, and T. De Bie, “Automatic chord estimation from audio: A review of the state of the art,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 556–575, 2014.

[8] R. Cohn, “Introduction to neo-riemannian theory: a survey and a historical perspective,” *Journal of Music Theory*, pp. 167–180, 1998.

[9] J. J. Fux and J. Edmunds, *The study of counterpoint from Johann Joseph Fux’s Gradus ad parnassum*. WW Norton & Company, 1965, no. 277.

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

[11] D. Huron, *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2008.

[12] T. Stapleford, “The harmony, melody, and form of herman, a real-time music generation system,” *Master’s thesis, University of Edinburgh*, 1998.

[13] B. Hughes, “Neo-riemannian triadic progressions,” *OPEN MUSIC THEORY*, 2021.