EvoComposer and DeepBach: Comparing approaches to the Four-Part Harmonization Problem

Jacob Helhoski

Department of Computer Science, Binghamton University

CS301: Ethical, Social, and Global Issues in Computing

Dr. George Weinschenk

December 5, 2022

Abstract

When solving the four-part harmonization problem, EvoComposer, a genetic algorithm, outperforms DeepBach, a neural network, with its competitive and restrictive approach resulting in compositions that better mimic human composition. DeepBach is a recurrent neural network, which uses a Markov-Chain-Monte-Carlo method with Gibbs-sampling. The creators of the method prove its value by presenting musical heuristics and data showing that professional musicians were unable to discern between real Bach chorales and DeepBach's generated chorales. EvoComposer is a genetic algorithm, which uses the NSGA-II method, representing beats of a chorale with chromosomes. EvoComposer mimics natural selection and stratifies its chorales in their evaluation, making the method both competitive and restrictive. DeepBach is less competitive and restrictive, since it does not produce chorales that compete with each other and does not gradually decrease its number of chorales. These two qualities prove that EvoComposer is more like human composition. Previous examples of successful algorithmic composition have relied on human methods. Because of this, EvoComposer has a greater potential to be combined with other fields of musical computing, such as music information retrieval.

EvoComposer and DeepBach: Comparing approaches to the Four-Part Harmonization Problem

While the battles for improved computing speed and cleaner graphical performance rage on, other algorithmic approaches, composing music in the spirit of Bach and Vivaldi, also compete for supremacy. Algorithmic composition, however, predates computing technology by over three hundred years, with the techniques of counterpoint and voice-leading defining the chorales of early western music. With the advent of computing technology, musically inclined programmers could apply these previously analytical rules to their work, creating novel approaches to composition. The keen insight into human musical perception and composition gained from this practice has led to a neo-Baroque period of research in the field of algorithmic music, with one recent algorithm showing promise in advancing the technology. When solving the four-part harmonization problem, EvoComposer, a genetic algorithm, outperforms DeepBach, a neural network, with its competitive and restrictive approach resulting in compositions that better mimic human composition. Through study of EvoComposer's compositions, the field of music theory can develop new methods of understanding and creating harmony. Before discussing EvoComposer's method, its method's similarities to human composition, and its advantage over competing algorithms, an understanding of the DeepBach method is necessary.

Alternative Technology

DeepBach, developed by Hadjeres, Pachet, and Nielsen in 2016, was called "steerable" by its creators (Hadjeres, Pachet, Nielsen, 2016), its potential for human interaction and constraints defining it as a true compositional approach, as opposed to a pure calculation. The authors also claim that DeepBach's use of Gibbs-sampling, a tool of convolutional neural networks, separates the algorithm from older, recurrent approaches (Wikipedia, 2022).

The algorithm first evaluates the randomized starting notes of a chorale independently of the composition's size or the notes' positions in the composition, as seen in Figure 1 (Hadjeres et al., 2017), allowing multiple transpositions of the same chorale (restricted to the vocal ranges of the four parts) to increase its data input. To process the classifications of prospective

Figure 1

DeepBach Representation of a Chorale

```
D5,__,E5,F5,D5,__,__,__,C5,__,__,__,E5
A4,__,__,__,G4,__,F4,__,E4,__,__,__,E4
C4,__,__,__,B3,__,__,__,G3,__,__,__,A3
F3,__,D3,__,G3,__,__,__,C2,__,__,__,C#2
 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1
 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0
```

compositions, each of these quality metrics uses three RNNs, which model the notes based on their place in a melody (same voice part, different time) or their resulting chord (different voice parts, same time), before funneling their outputs into a final RNN (Hadjeres, Pachet, Nielsen, 2016). The cycle continues, as a Markov-Chain Monte Carlo (MCMC) procedure samples from the data. Distributions from this method are irreversible, such that equations creating new chorales based on sampled data, normally used in calculating a Markov-chain's stationary distribution, are incompatible with the data. According to the authors, this characteristic works to the benefit of the algorithm, since a method, which finds the Markov-chain's stationary distribution through direct calculation, still does exist, outperforming reversible methods at sampling from the data.

The authors conducted a study of their algorithm, focusing less on how DeepBach compares to other approaches, and more on how the algorithm performs at increasing levels of complexity. Human participants, with varying degrees of musical experience, confirmed the authors' belief that the algorithm could produce chorales indistinguishable from Bach's own work. These results may convince a listener of DeepBach's ability to mimic Bach's style, but they beg further discussion of aesthetic considerations. An overview of EvoComposer's compositional method, a method that considers this factor, begins to demonstrate its advantage over DeepBach.

Support

Technical Details

Introduced in 2020, EvoComposer applies an evolutionary algorithm to this reharmonization problem (De Prisco et al., 1). Evolutionary algorithms optimize systems for multiple objectives by representing data as chromosomes. EvoComposer defines chorales in this manner, with each beat being a gene (*Figure 2*) (DePrisco et al., 2020), optimizing the chorales to reach mathematically defined melodic and harmonic objectives.

Figure 2

Genes in the EvoComposer Model

The authors' method, following the NSGA-II strategy, proceeds as follows: First, generate a population of random chorales with notes in the starting key, initializing them with values for fitness. Then, sort the chorales by fitness, then use tournament selection to extract a second population. Next, modify the second population using crossover and mutation, before mating it with the original population. Finally, determine a new population from sorting the resulting chorales by statistical density, repeating the entire process with this new population until the algorithm is stopped. The authors maintain that the principles of elitism, the preservation of certain musical traits, and diversity, the generation of new melodic and harmonic ideas, are beneficial consequences of using this evolutionary algorithm. Giving rise to these principles are the driving forces of competition and restriction.

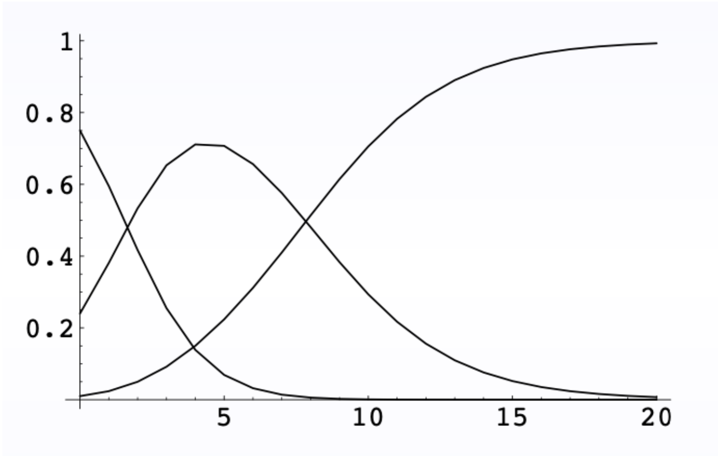
Chorales generated by EvoComposer face layers of testing in the algorithm, such as melodic and harmonic objectives, and the final numerical tests of compositional value and creativity. They compete for stylistic accuracy and musical creativity. The paper published by Deb et al., which outlines the NSGA-II algorithm, highlights how the algorithm's potential solutions "compete with their crowding distance" (186). Solutions are effectively encouraged by the algorithm to pass on traits to their offspring that would set these offspring apart from statistically dense areas of a plot. Other methods that are non-competitive can also be used to generate new chorales.

DeepBach samples the best chorale generated in each iteration, similar to EvoCompser, but the driving force behind the development of these chorales is ultimately randomness. Its method of pseudo-Gibbs sampling (Hadjeres et al., 5), repeatedly samples from the potential chorales that could be derived from the current. Doing so generates chorales in a way that doesn't compare multiple against each other. The DeepBach chorales are generated to be as valuable as possible, but not better than any other compositions. DeepBach's chorale generation is, as a result, non-competitive in nature.

EvoComposer also utilizes a restrictive method, specifically in its evaluation of chorales, meaning that many randomized chorales of roughly equal quality are stratified into different levels of quality over iterations of the algorithm. As the algorithm progresses, a few select chorales reach the desired quality, restricting the size of a large group. Jonathan Rowe's presentation summarizes much of the mathematical theory underpinning the function of genetic algorithms, corroborating the idea of restriction. A distribution produced by the algorithm, he says, will eventually converge (*Figure 3*) (Rowe, 2012), where the highest sorted chorales will share characteristics among themselves (Rowe, 2012), differing from the random characteristics of the starting group.

Figure 3

Converging Population of Solutions in Genetic Algorithm Model



DeepBach's method of restriction differs from EvoComposer's method. When determining the next chorale, the algorithm generates numerous potential replacement chorales, sampling one for the next iteration. DeepBach restricts possibilities, but this restriction happens in an instant, multiple times, whereas EvoComposer restricts its chorales gradually, with the entire algorithm being a single restriction. DeepBach could be better described as gradual modification of a single chorale.

The comparison of these two algorithms goes beyond their technical specifications. Their uses and connections to other fields warrant discussion as well.

Social Impact

Defining the role competition and restriction play in EvoComposer, on its own, does not prove EvoComposer's advantage in the study of music, but the algorithm's connection to human composition and adjacent fields of music computing do prove this advantage.

During their creative process, a human composer embodies the ideas of competition and restriction. Constantly attempting to improve upon a previous idea, he develops current compositions, combining the best features from multiple compositions he may have lingering in his mind. By creating unique musical ideas, he improves the quality of his work over his previous work. In this way, he competes against himself. While this process occurs, a restrictive process also influences the choices he makes in his composition. There exist many musical ideas in the mind of a composer, relating to either the particular piece that they are working on or embodying a concept to abstract to apply to just one piece. As the current piece develops, these ideas become refined, concrete melodic or harmonic elements, finding a place in the structure of the piece. Though creation is happening, it happens in a restrictive way, where the composer must intentionally limit himself for the sake of his work.

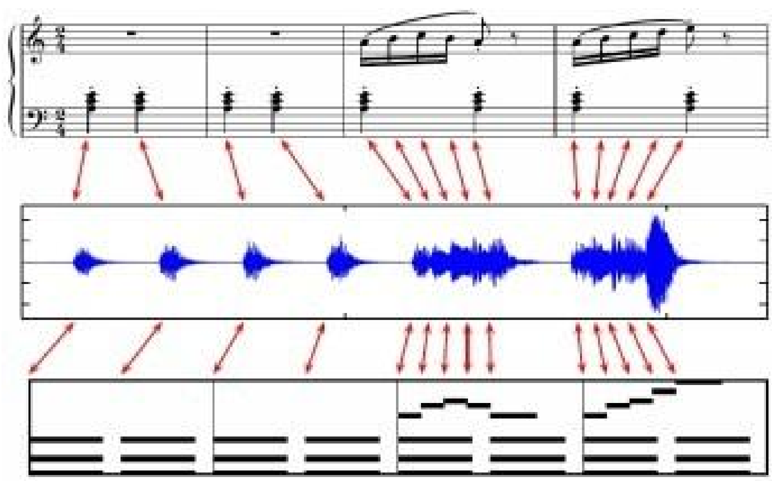
In his talk for a TED conference, Pierre Barreau, self described as a developer and a lover of music, introduces his own algorithmic composition software, named AIVA. He relates his software's methodology to that of humans, who engage in a "trial-and-error process, during which we might not get the right notes all the time,..correct[ing] ourselves" (Barreau, 1:30). This methodology acknowledges how ideas will develop and and be corrected during composition, incorporating the principles of competition and restriction. A large, professional orchestra recorded one of AIVA's generated pieces, which was well-received by the crowd at the conference (Barreau, 3:35), proving AIVA to be a quality example of algorithmic composition. It should follow, then, that EvoComposer, its driving forces being nearly identical to Barreua's software, belongs to a class of advanced music composition software inspired by human composition.

EvoComposer, and algorithmic composition, refrain from being solitary technologies, having uses in other subfields of music computation. This is not surprising, as evolutionary algorithms, one of which defines the functionality of EvoComposer, have had success in other creative problems (De Prisco et al., 1). One particular subfield, MIR (music information retrieval), does not create music, but instead uses algorithms and software to extract information from a vast collection of music.

In their article discussing MIR's analysis of salsa music, Arce-Lopera and Sarria M. mention how musical ideas like form and repeating patterns can be determined from a recording (59), a stark difference from algorithmic composition, which most always requires a simplification of the piece under examination, be it a MIDI arrangement or some other model (*Figure 4*) (Müller, 2008) . In this example, an opportunity presents itself for EvoComposer to be combined with MIR technology. A piece could be interpreted in real time by MIR methods,

Figure 4

Translation Between Different Musical Representations using MIR



while a more advanced version of EvoComposer generates and optimizes a population of compositions based on the piece. A third technology could simply test the highest ranked composition against the real recording, gaining insight into how the piece either adheres to or subverts what the algorithm believes it could or should do. This being only one example of algorithmic composition's potential synergy with other musical technologies, it is apparent that much more research is needed to be done into algorithms like EvoComposer. Current research into algorithmic composition, however, proves EvoComposer as a stronger framework for composition, with the principles of restriction and competition also defining its advantage as an augmentation of other fields.

Conclusion

The discussion of differences in the technical and social aspects of the DeepBach and EvoComposer approaches to algorithmic composition provide important insights regarding the role the field will play in our understanding of music. The force of competition drives EvoComposer's generation, much like other genetic algorithms, while DeepBach creates non-competitively. A lack of restriction in DeepBach's system of evaluation differs from EvoComposer's restriction of its domain, once again setting the two methods apart. A human composer constantly relies on these forces while composing, restricting a group of potential melodies or harmonic structures while determining which of them he deems better than the others. This understanding of the composing process shows that EvoCompser generates chorales far more convincingly than DeepBach, proving its higher quality and similarity to human composition.

If a professional in the field of music information retrieval must decide between the two methods, EvoComposer should be chosen, since algorithms that produce human results are more well suited to these problems, and they see more frequent use in these problems, as evidenced by the example of the AIVA algorithm. With these new insights, professionals in the computational fields of music analysis, MIR, and algorithmic composing can hopefully discover and develop the synergy between these different fields, helping to make the "neo-Baroque" period into a true musical renaissance.

References

Arce-Lopera C. & Sarria M. G. (2020, November). Understanding salsa: how computing is defining latin music. *Communications of the ACM*, *63* (11), 59-60. https://doi.org/10.1145/3416967

Deb. K., Pratap A., Agarwal S., & Meyarivan T.. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197. https://ieeexplore.ieee.org/document/996017

De Prisco, R., Zaccagnino, G., & Zaccagnino, R. (2020). EvoComposer: an evolutionary algorithm for 4-voice music compositions. *Evolutionary Computation, 28*(3), 489-530. https://doi.org/10.1162/evco_a_00265

Gibbs sampling. (2022, May 4). In *Wikipedia*. https://en.wikipedia.org/wiki/Gibbs_sampling

Müller M. (2008). [*Sheet music translated to MIDI]* [Image]. Max-Planck-Institut-Informatik. https://resources.mpi-inf.mpg.de/departments/d4/teaching/ws200809/mir_mm/index.html

Hadjeres, G., Pachet, F., & Nielsen, F. (2017). DeepBach: a steerable model for Bach chorales generation. *Proceedings of the 34th International Conference on Machine Learning, 70,* (pp.1362-1371). https://dl.acm.org/doi/10.5555/3305381.3305522

Rowe, J. E. (2012). Genetic algorithm theory. *GECCO '12: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation,* (pp.917-940). https://doi.org/10.1145/2330784.2330923

TED. (2018, October 1). *How AI could compose a personalized soundtrack to your life | Pierre Barreau.* [Video]. YouTube. https://www.youtube.com/watch?v=wYb3Wimn01s