STEM Fellowship Big Data Challenge 2018-2019

Crowdfunding Capital: A Study on Factors Related to Kickstarter Success

Aytug Altin

Webber Academy

**Both authors contributed equally to the writing and research in this study. Their names are listed in alphabetical order.*

June 28, 2021

Abstract

19 abstract **Keywords** Keyword-Keyword; Keyword; Keyword Keyword; Keyword-Keyword Keyword

1 Introduction

Evolutionary algorithms can be used for both optimization problems and modelling problems [1]. In both cases, we are looking for some input that creates a known or desired output. Modelling problems can be transformed to optimization problems where our search space is defined by all the potential models.

In the domain of music composition we lack a model that could judge the input, the fitness function, and our desired output may be ill-defined. In GenJam [3], the quality of the genetically produced solos are rated by human individuals. A human mentor gives "real-time feedback" which is used to derive a fitness score. Fitness can also be calculated by different aspects based on music theory as shown in [5], [2], [4]. By splitting the fitness function into subcategories, we are allowed to rate an individual song based on different aspects and set importance to certain preferred aspect. In [2], songs were compared to a well known group of songs. In [5], each bar of the song is rated by different criteria and summed up together to obtain the total fitness. The fitness is here calculated by the similarity between the to be rated individual and a reference individual or by reference values. Five fitness functions where used in [4], one per type of user preference. These preferences are: transition, repetition, variety, range and mood.

In this paper, we emphasized on the structure of the individual songs. Instead of focusing on theories that define music to be better than others, we focussed on a master song that defines the theory similar to [5]. The similarities between the population and the master can be based on the result of an absolute comparison i.e. comparing the exact number of notes of the candidate, which is the to be rated song, to the master. However, these absolute ratings would result in a population that is exactly the same as the master song, this is not what we are looking for. We introduce a new way to rate the candidates: relative ratings.

2 Model

In this section, the complete model is discussed. On figure 1 an overview of our model is illustrated. Before we go into detail, we briefly discuss the structure of our model. The initial population can consist of both random songs or non-random songs, these are called $GEN0$. The next generation is calculated by applying a crossover function on $GEN0$ to obtain $GEN1$ which is the offspring. The offspring is calculated by using a crossover function. Now we select the best rated songs in $GEN1$ and mutate them. This process is repeated until a certain condition is met or un-

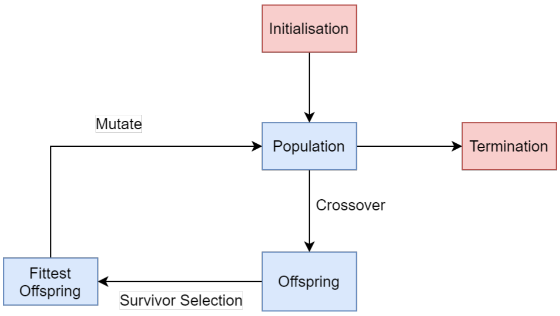

Figure 1: The genetic model, an overview of the important components.

til user is satisfied with the results and manually terminates the loop.

2.1 The genetic algorithm

First the initialisation process will start. During the **initialisation** the the songs are loaded or created and put into the initial population. The initial population usually consist of a mix of different songs, here we want to make sure we have a diverse population. The size of the population which is static during the execution is decided here. The whole generation algorithm will be configured during this process, things such as filling in parameters also done here.

2.2 Crossover

The system performs an uniform crossover. We consider the notes, chords and rests as the alleles that are selected in the offspring. Two parents produce one child and all parents are paired with each other. We get for N parents $\frac{N*(N-1)}{2}$ children.

On figure 2 two parents are graphically displayed. During the crossover, we iteratively go over each offset an consider the alleles of both parents on that offset to be selected. Both alleles have an equal chance to be selected. On figure 3 a sample child of the corresponding parents is illustrated.

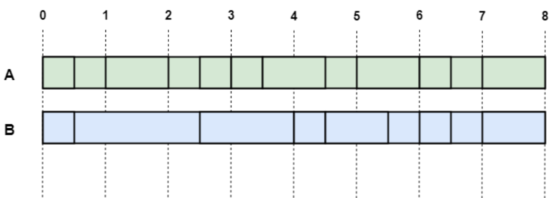

Figure 2: Two parents A and B. Each rectangle is considered to be a note, chord or rest with its own size.

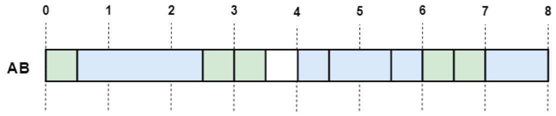

Figure 3: Offspring of the parents A and B.

This process is repeated multiple times during the execution, every song in a group must be paired with every other song. So we get for N parents $\frac{N*(N-1)}{2}$ children.

2.3 Survivor selection

test

2.4 The fitness function

A composition has multiple aspects that can be rated at an individual level as it should. Therefore, the fitness function can be divided into multiple sub-functions that belong to a concept. This concept can be seen as what it is that the function-gives a score, e.g. the tendency to follow an musical scale within the composition can be a concept. Every sub-function calculates a score and to this score there is a predetermined weight attached that implies the importance of the rated concept. The total fitness of a song x is equal to the sum of products of all sub-functions S an their corresponding weights.

$$TotalFitness(x) = \sum_{S=1}^{n} S_{concept} * W_{concept}$$

$$S(x) = difference(f(x), optimalscore) * S_{weight}$$

2.4.1 The master song

The fitness function calculates a score song based on the master. The master song is set during the initialisation process and it controls the population by defining the rules. We can rate candidate songs based on the master song in two ways:

- absolute comparison: this is where we compare the elements of the master song directly with the candidate song.
- relative comparison: this is where we compare the relative structure of the master song directly with the candidate song.

Materials & Methods

man

Results

results

Discussion

Discussion

Conclusions

concluded

Acknowledgements

We would like to extend our sincerest gratitude and appreciation to everyone who helped make this project a possibility.Acknowledgements Acknowledgements Acknowledgements Acknowledgements Acknowledgements Acknowledgements Acknowledgements Acknowledgements

References

[1] James E. Smith A. E. Eiben. *Introduction to Evolutionary Computing*. Mairdumont Gmbh & Co. Kg, 2003.

[2] Viktor. Anderling, Olle. Andreasson, Christoffer. Olsson, Sean. Pavlov, Christian. Svensson, and Johannes Wikner. Generation of music through genetic algorithms. Master's thesis, Chalmers University of Technology University of Gothenburg Department of Computer Science and Engineering, Gothenburg, Sweden, 6 2014.

[3] John Biles. Genjam: A genetic algorithm for generating jazz solos. 07 1994.

[4] Adil Haleem Khan. Artificial intelligence approaches to music composition. Northern Kentucky University, Highland Heights, KY 41099, 2013.

[5] Dragan Matić. A genetic algorithm for composing music. *Yugoslav Journal of Operations Research*, 20, 01 2010.