

Assignment task:

Generate an accompaniment for a given monophonic melody using genetic algorithms. The accompaniment consists of a set of chords and each chord consists of three notes.

Algorithm flow:

Input:

The program take input MIDI file (*.mid) that has monophonic melody, analyzes it and get the detected key and it's scale. Then, analyzes the average note of each quarter of a bar and passes these information to the genetic algorithm.

Genetic Algorithm components:

Genetic Algorithm representation:

- Population is a set of individuals
- Individual is a list of chords, representing a solution
- Chord consists of 3 notes
- Note is a number representing a midi value, this value is determined using an octave and note.

Population size and number of generations:

- Population size and number of generations are defined at the beginning of the main function. The size of the population and number of generations were calculated fittest through experimentation so it can give best results with respect to time taken.

Selection technique:

- The selection technique used is the ranking selection, first the fitness value is calculated for the whole new population then only the fittest half of the population is selected.

Fitness function:

- The individual's fitness value depends on two things:
 - The similarity of each chord with the average note played on the corresponding quarter of a bar.

- The existence of this chord in the scale of the detected key of the melody.
- The weights of these functions were determined by experimentation.

Mutation technique:

- I combined both inversion with the bit-flip mutation techniques.
- The probability of each individual to get a mutation is 15%.
- The mutated individual got its chords reversed and one chord changed randomly.

Crossover technique:

- The crossover technique used is N point crossover where N is the length of a single individual.
- Each child is the exact opposite of the other.

Flow:

- After analyzing the input, the program start with creating the first generation using individuals of same size and random chords.
- The created population goes under selection process and only half of the population is selected.
- The selected population goes into crossover to produce the other half of the population.
- This continues for N generations, then the program calculates the fittest individual among all generations, and it would be our accompaniment for the given melody.

Dependencies:

- Python 3.8
- Python library mido
- Python library music21
- Python library numpy and random

Key identification:

- I used a python library(music21) to parse the input and get the identified key for the melody.

Program running Manual:

- Add a call to the main function with the path to the input melody (*.mid) file and path to the output melody (*.mid) file. Then, you can run the code safely.
- It will take ~(1 - 3) minutes for each input file.

Keys detected for the input files:

- input1.mid file -> d minor
- input2.mid file -> F major
- input3.mid file -> e minor