

Introduction to AI (Assignment 2)

Description

Generate an accompaniment for melody using an evolutionary algorithm.

Accompaniment is represented by a sequence of chords. Each chord contains exactly three notes. Evolutionary algorithm considers only the next types of chords:

- major and minor triads
- first and second inversions of major and minor triads
- diminished chords (DIM)
- suspended second chords (SUS2)
- suspended fourth chords (SUS4)

How to start

Install libraries

You must install the following libraries before start:

- `mido`
- `numpy`

Start program

- You can launch the code using console.

Note: you should be at the same path with file `AndreyKuzmickiy.py`. Be sure that input files exist.

```
python AndreyKuzmickiy.py input1.mid
```

- If you are using **IDE**, e.g. *PyCharm*, then just open the file: `AndreyKuzmickiy.py` and launch it using IDE tool.

Note: you should configure environment to launch program with input parameter (e.g. input1.mid)

Output file creation

Output file will be located in **output/** directory. If this directory does not exist, then the program will create it. The full relative path of the output file will be: */output/param*, where *param* is input parameter for the program.

For example, for the following command: `python AndreyKuzmickiy.py input/input1.mid` the following directory will be created: `/output/input/input1.mid`. The following output file will contain initial melody with accompaniment.

Algorithm flow

You are allowed to change the following parameters (in the program file):

- Number of generations (`GENERATIONS`),
- Size of each population (`POPULATION_SIZE`),
- Number of generated accompaniments used in crossover (`CROSSOVER_NUMBER`),
- Chance for mutation of accompaniment (`MUTATION_CHANCE`) should be < 1 ,
- Chance for mutation of a chord in accompaniment (`MUTATION_ACCORD_CHANCE`) should be < 1 ,
- Chance to make chord empty (not play) in accompaniment (`MUTATION_EMPTY_ACCORD_CHANCE`) should be < 1 .

Steps of the algorithm

1. The program converts input MIDI file to matrix represented start and duration of each chord.
2. Call evolutionary algorithm with input matrix (data). On each step:
 - a. Crossover function called for 2 randomly chosen accompaniment in population.
 - b. Mutation (with given probability) changes population.
 - c. Newly generated population appends to the best accompaniments, calculates fitness for each accompaniment for the population, sorts it and new best

population of given size is saved (another part is discarded).

3. From the population the program takes the best accompaniment (in terms of fitness function) and converts it to MIDI file.
4. Concatenate both initial melody and generated accompaniment and creates the output MIDI file.

Crossover

Takes 2 accompaniments and changes their melody. Takes the first accompaniment and moves its chords to chords (of the same starting time) of the second accompaniment with some speed depending on distance between the chords.

Mutation

Mutation can randomly change a chord of an accompaniment with given probability. Also, it can omit accord with given probability.

Fitness function

Fitness function is sum of criteria for an accompaniment depending on the given melody. Each criterion has its own cost on which it will be multiplied. Fitness function has the following criteria:

- Key note of the accompaniment should be less than key note of the initial melody.
- The distance between chords of the accompaniment should be minimal as much as possible for good melody.
- The shift (distance between current and next chords) should not be large.
- Accompaniment should not have many omitted chords (holes).
- Chords of the accompaniment should have the same key as music.
- Chords of the accompaniment should be placed according to tonic of the key in melody.