Automatic Trance Music Generation and Automatic Melodic Variation

as Aesthetic Music?

Marvin Beese

University of Potsdam

Outline

Introduction

- generation of music in AI
  - musical rules
  - important for aesthetic understanding
- generation of Trance music with Armin
  - Computer-aided Composition tool
  - melodic and rhythmic implementation using ASP
- automatic melodic variation with AIM
  - musical alteration of a given melody with ASP

Aesthetics of generated music: Musical Aesthetics

- aesthetic judgement:
  - finding elegance and novelty
  - minimization of incoherence and boredom
- tonal and rhythmic coherence
- stability of melodic contour
- balance of repetition and variety

Aesthetics of generated music: Genetic Algorithms

- origin: evolutionary biology
- new elements: combination of elements from successful outcomes
- mutation: changes for improvement
  - variation of melody
- evolutionary mutation: mutation for every generation
  - change of pitch for random note
  - split & merge for musical sections

Aesthetics of generated music: Comparison

Table. An overview of the results of assessment by aesthetic judgement.

Beginning state	Unprocessed	Fittest unmodified	Evolutionary mutations	Musical mutations	Combined mutations
Random	★	★	★	★★	★
Rule-based	★★★	★★★	★★	★★★★	★★★★

- comparable results for rule-based & mutated music
- well formed melodies
- evolutionary melody:
  - reduction of coherence of form
  - less gain in novelty
- $\rightarrow$ very little advantage for evolutionary music

Armin: Trance Music Composition: Computer-aided Composition

- computer-aided composition
  - algorithmic composition
    - ★ next note?
    - ★ duration of the note?
- rules for the music-genre

Armin: Trance Music Composition: Background of Trance Music

- electronic dance music with 130-140bpm
- time signature $\frac{4}{4}$
  - kick on the second beat
  - snare drum/clap on the fourth beat
- change of pace every second/fourth/eighth bar
  - with change of drum/instrument
  - $\rightarrow$ progression
- breakdown: longer synthetic chords, slower pace

Armin: Trance Music Composition: Background of Trance Music

- based on Anton
  - harmonic, melodic, rhythmic composition system
- musical sections chaining
  - e.g. intro→verse, verse→chorus, chorus→breakdown, ...

Armin: Trance Music Composition: Assembler File

- *arminAssembler.lp*: definition of order and frequency of parts
- model over timestep T
  - play the intro

```
playState(intro,1) :- part(intro).
```

  - section for following timestep

```
1 { playState(verse,T+1),
playState(chorus,T+1),
playState(breakDown,T+1) } 1 :- playState(P,T), timeScore(T),
                                statesNumber(SN), T < SN-2.
```

  - play the outro

```
playState(outro, SN) :- part(outro), statesNumber(SN).
```

Assembler File

- model over timestep T
  - no consecutive verses
    :- playState(verse, T), playState(verse,T+1).
  - no three consecutive played parts
    :- playState(P,T), playState(P,T+1), playState(P,T+2).

Armin: Trance Music Composition: Rhythmic Component

- time signature $\frac{4}{4}$: 32 pulses
  pulseMeasureLimit(32).
- whole note: 32 pulses, half note: 16 pulses
  longDurations(16;32).
- melody contains 8 measures
  lastMeasure(8).

Rhythmic Component: Example

- Example for an 8 bar configuration

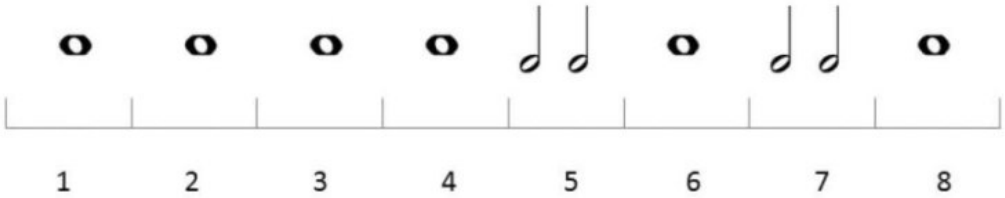

Figure 5. Example of an 8 bar configuration

- a half or whole note can follow a whole note

```
1 { durationMeasure(0,D1,M+1,C+1)
        : longDurations(D1) } 1 :- durationMeasure(0,DR,M,C),
                                   DR == 32, lastMeasure(LM), M + 1 <= LM.
```

- half notes must come in a pair

```
1 { durationMeasure(16,D1,M,C+1)
        : longDurations(D1) } 1 :- durationMeasure(0,DR,M,C), DR == 16.
```

- a half or whole note can follow two half notes

```
1 { durationMeasure(0,DR,M+1,C+1) : longDurations(DR) } 1 :- durationMeasure(16,16,M,C),
                                   durationMeasure(0,16,M,C-1), lastMeasure(LM), M + 1 <= LM.
```

Melodic Variation with ASP: Background

- what notes should be preserved? / what notes can change?
- Alterations in Music (AIM)
  - based on Anton
  - rhythmic characteristics like in Armin
- input file with specifications
  - maximum value of notes to be changed
  - chosen note
  - duration of the note
    see the following example of the input file

Melodic Variation with ASP: Input File

```
partTimeMax(P,5).

%%Change notes
numberOfNotesToChange(1).
toChangeNumber(1..CN) :- numberOfNotesToChange(CN).

%% choosenNote(part, note, counter).
choosenNote(1,25,1).
choosenNote(1,24,2).
choosenNote(1,22,3).
choosenNote(1,20,4).
choosenNote(1,22,5).

%% duration(start, duration, measure, counter).
duration(1,16,1,1).
duration(16,8,1,2).
duration(24,8,1,3).
duration(1,16,2,4).
duration(16,16,2,5).
```

Figure 2 Fragment of code of the input file

Melodic Variation with ASP: Melodic Variations Engine

- which notes keep the essence (and will not be changed)?
  - first note: indicates fundamental
  - second note: keeps progression of the first note
  - last note: preserves ending of the melodic line

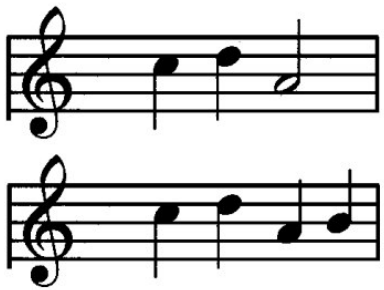

Figure 5 Note conversion. Split a note into halves

- changing a note when choosen by
  - changing the pitch
  - splitting the note into equivalent halves
    - ★ first note remains, second note changes

Conclusion

- automatic generation of music:
  - rule-based
  - evolutionary methods
  - $\rightarrow$ aesthetics of music?
    - ★ rule-based: valid music, less novelty
    - ★ with mutated iterations: not too much improvement

Conclusion

- Armin
  - noteworthy expansion of Anton with Trance genre
  - section chaining
  - rhythmic focus
- AIM
  - not a wide variety of musical alterations
    - ★ freedom of change for nearly every tone (regardless musical criteria)
  - multiple iterations: works like evolutionary musical mutation

Sources

- Georg Boenn, Martin Brain, Marina De Vos, John P. Fitch: Automatic music composition using answer set programming. TPLP 11(2-3): 397-427 (2011)
- Andrew R. Brown: An aesthetic comparison of rule-based and genetic algorithms for generating melodies (2004)
- Flavio Omar Everardo Pérez, Fernando Antonio Aguilera Ramírez: Armin: Automatic Trance Music Composition using Answer Set Programming. Fundamenta Informaticae 113 (2011) 79–96
- Flavio Omar Everardo Pérez: A Logical Approach for Melodic Variations. LA-NMR (2011)
- Link for Trance Music by Flavio Everardo

Thank you for your attention!