ALGORITHMIC MUSIC COMPOSITION

Jan Dreier

June 19, 2015

RWTH Aachen University

HISTORY

Mozart's dice game

”Anleitung zum Componieren von Walzern so viele man will vermittelst zweier Würfel, ohne etwas von der Musik oder Composition zu verstehen”

=> 45,949,729,863,572,161 different yet similar waltzes

Source: https://www.youtube.com/watch?v=fK2MCXpDWB4

1955 Hiller, Isaacson: First computer-generated composition

1991 Gibson, Byrne: Musical Composition Using Genetic Algorithms And Neural Networks

2011 Donnelly, Sheppard: Evolving Four-Part Harmony Using Genetic Algorithms

MUSIC COMPOSITION AS SEARCH PROBLEM

MUSIC COMPOSITION AS SEARCH PROBLEM

Goal:

- Compose music without or with minimal human guidance

Approach:

- Search the set of all possible compositions
- Return one which sounds good

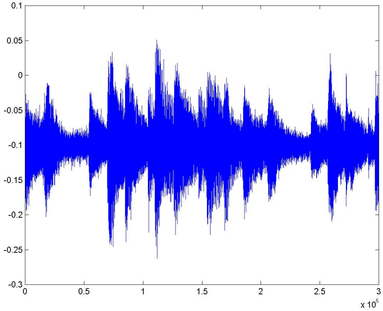
Problems:

- How to represent music?
- Search space is very big – how to search efficiently?
- How to evaluate if something sounds good?

How can we represent music?

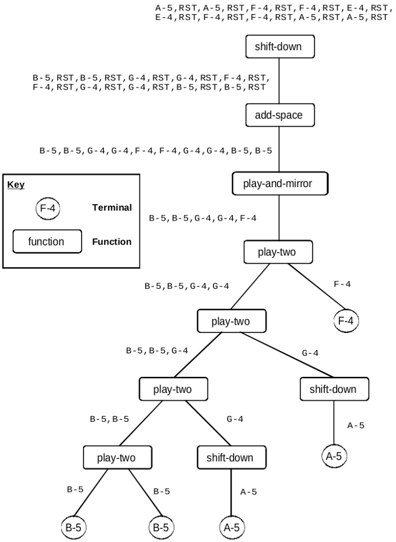
- Audio files
- Flat structure
- Hierarchical structure

MUSIC REPRESENTATION



Source: http://www.angelfire.com/art2/speech-audio-seperat/

Source: http://www.well.com/user/bryan/last.gif



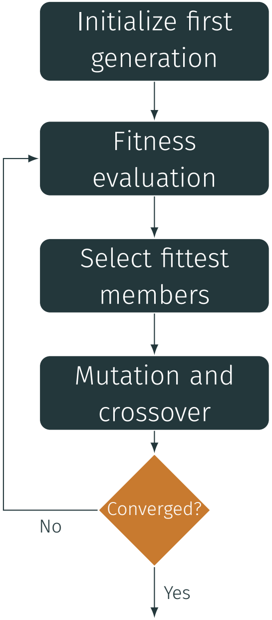
Source: http://graphics.stanford.edu/ bjohanso/papers/gp98/johanson98gpmusic.pdf

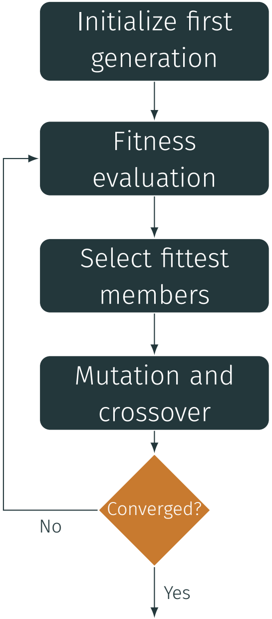
EFFICIENT SEARCHING

How can we search efficiently?

- Genetic algorithms

GENETIC ALGORITHMS





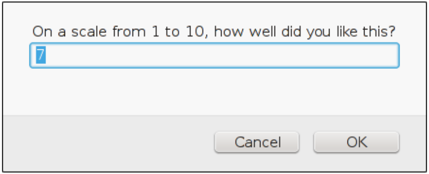
Example: Maximize function $f : \mathbb{R}^n \mapsto \mathbb{R}$

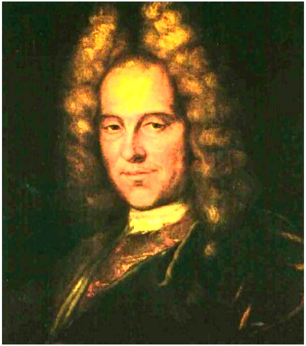
- **Initialization:** Select $x_1, \ldots x_k \in \mathbb{R}^n$ at random
- **Fitness** of x: $f(x)$
- **Mutation:** Shift by random $\varepsilon \in \mathbb{R}^n$
- **Crossover** of x and y: Choose value from line-segment between x and y

MUSIC EVALUATION

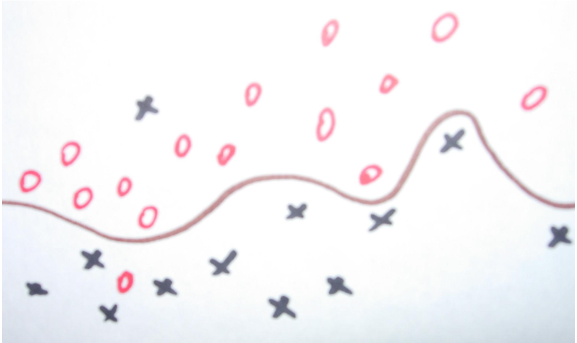
How can we evaluate music?

- Human based
- Rule based
- Machine learning based


On a scale from 1 to 10, how well did you like this?
7
Cancel
OK



Source: https://en.wikipedia.org/wiki/Johann_Joseph_Fux



Source: http://conferences.telecom-bretagne.eu/fps2012/program/slides/07.pdf

EVOLVING FOUR-PART HARMONY ...

Patrick Donnelly, John Sheppard
Evolving Four-Part Harmony Using Genetic Algorithms
Applications of Evolutionary Computation, 2011, pp 273-282

Used techniques:

- Flat music representation
- Genetic algorithms
- Rule based Fitness

Music representation:

- Four parallel parts
- Each part: List of (pitch, duration) tuples
- In C-major
- Total length not fixed

Example:

```
part 1: { (60, 1/4), (62, 1/4),
          (64, 1/4), (62, 1/4) }

part 2: { (48, 1/2), (52, 1/2) }

part 3: { (52, 1/2), (55, 1/2) }

part 4: { (55, 1/4), (57, 1/4),
          (59, 1/2) }
```

Genetic Algorithms

Initialize first generation with C-major chords

Mutation: Apply one of these operations to a random part

Repeat Note:

→

Shift Random Note:

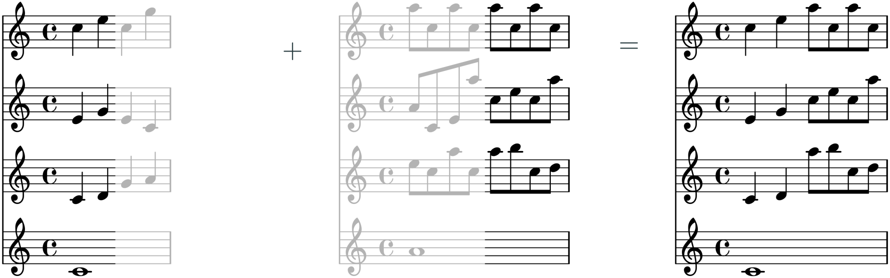
→

Alter Length:

→

Total: 11 operations

Crossover: Cut and glue two elements together



Rule Based Fitness

RULE BASED FITNESS

What does good music need?

- Melody (Should be catchy)
- Harmony (Melodies should interact nicely)
- Rhythm (Emphasize meter)
- Structure (Intro, outro, reoccurring themes, ...)
- Timbre/Intonation

$\Rightarrow$ Music theory provides tools to enforce these constraints

RULES

- **Leap Height:** Two consecutive notes should not have an interval larger than a 9th
- **Voice Crossing:** An upper part should always play higher than a lower part
- **Opening/Closing Chord:** The piece should start and end with a C-major chord
- **Intervals:** Pure and dissonant intervals should be avoided
- **Total:** 15 rules

$\text{fitness}(\text{rule}_i) = \frac{n_i - v_i}{n_i}$

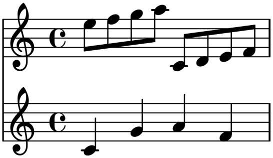
total fitness = $\sum_i (\omega_i \cdot \text{fitness}(\text{rule}_i))$

- n_i = Number of places where rule i could be violated
- v_i = Number of places where rule i actually is violated

EXAMPLE

$\text{fitness}(\text{rule}_i) = \frac{n_i - v_i}{n_i}$

- n_i = Number of places where rule i could be violated
- v_i = Number of places where rule i actually is violated



Leap Height:

- n_i = number of leaps = 10
- v_i = number of large leaps = 1
- fitness(Leap Hight) = $\frac{9}{10}$

Voice Crossing:

- n_i = num. of parallel notes = 4
- v_i = number of crossings = 2
- fitness(Voice Crossing) = $\frac{1}{2}$

Pro:

- Strong music-theoretical foundation
- Works well in practice

Contra:

- Humans need to define the rules for each genre (Cannot be automated)
- Some genres are hard to express by rules

SUMMARY

We discussed:

- Genetic algorithms
- Rule based fitness functions

Using these tools it is possible to generate interesting music without human interaction.

SOURCES

Papers:

- Patrick Donnelly, John Sheppard
  Evolving Four-Part Harmony Using Genetic Algorithms
  Applications of Evolutionary Computation, 2011, pp 273-282

Images:

- https://en.wikipedia.org/wiki/Johann_Joseph_Fux
- http://conferences.telecom-bretagne.eu/fps2012/program/slides/07.pdf
- http://www.angelfire.com/art2/speech-audio-seperat/
- http://www.well.com/user/bryan/last.gif
- http://graphics.stanford.edu/ bjohanso/papers/gp98/johanson98gpmusic.pdf
- https://www.youtube.com/watch?v=fK2MCXpDWB4

END