Paper Review of "The Use of Constraint System for Musical Composition"

Geraint A. Wiggins
Department of Artificial Intelligence
University of Edinburgh
Reviewer : ChangHyun Kim

USC ISE 575c, 2007

Index

Introduction

- Covers three issues, "Four-part harmony; serial composition; and music representation", which have been largely ignored by the current research
- In section 2, Genetic Algorithm for four part harmonisation
- In section 3, Simple CLP(Constraint Logic Programming) based system
- In section 4, Some subtleties of the common notation for pitch

Problems using Constraints to Harmonise

- Genetic Algorithms(GAs) to music
- Sommuk Phon-Amnuaisuk, a Ph.D student at Edinburgh
- Why GA system in this paper?
  - Because, from one point of view, a GA is very much a constratint satisfaction mechanism by fitness function.

What is the fitness function?

- The fitness function judges the fitness of each chromosome(key or chord) according to the following criteria derived directly from music theory.
- Criteria : We avoid parallel unison, parallel perfect 5ths, and parallel octaves.
- Penalization : Solutions are penalised for note doubling and omission, in the primary major and minor triads.

The Outline of a GA

- Derived from the Theory of Natural Selection of Darwin
- Chromosomes, a potential solution to the problem we are trying to solve – Homophonic harmonisation of a monophonic melody
- Two operators : mutate and crossover
- Fitness function : measure of how good a solution any given chromosome is
- GAs are a stochastic “search method” and they are generally regarded as a “weak” method in that they are not problem-specific.
- Problem specific mutations makes a huge increase in the effectiveness of GA search. Even in spite of the obvious problems with stochastic methods, GAs have been very succesful in difficult problems such as timetabling and scheduling.

GAs as Constraint Solution Mechanisms

- Each chromosome is a potential solution which conforms to constraints specified in the fitness function to a greater or lesser extent.
- Rules of the fitness function : Solutions(chromosomes) which does not conform to constraints, die out of the chromosome pool, so search is restricted as time proceeds to sets of potential solutions which more and more conform to the constraints. (“among the adjustable chromosomes to the fitness function”)
- In this paper, directed mutations are used to apply musically meaningful mutations to chromosomes, and so cause changes which are known to be beneficial, which helps to improve the chromosomes’ conformity to the required constraints.

Case Study: Four-Part Harmonisation

- Problem : Four part vocal harmonisation.
- Harmony : 4 parts have n notes(chromosomes in GA) each, First row being the melody, population was initialised randomly, Fitness function encoded some basic rules of harmony.
- Search space for solutions to most harmonisation problems is enormously convoluted.
- High values are bad, so the high, jagged lines at the back of the graph correspond with the random initialization, while the lower, smoother parts at the front corresponse with later, more evolved generations. However, the fitness profile does not flatten out completely: There are isolated spikes of “badness”. ?

Fitness Profile for harmonised chromosome, over time

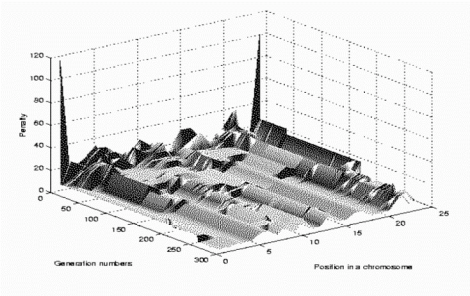


Figure 1: Fitness profile for harmonised chromosome, over time

Discussion

- Four-part harmonisation problem, where one is working on the level of individual voices, rather than on chords, it is rarely possible to change a note without having a knock-on effect on at least some, if not many, of the notes around.
- Jumping from one such basin to another may involve changing many details of a chromosome at once. Standard GAs are not good at doing this, and even if they were, the chances of all the necessary mutations being randomly applied at once are very small.
- But the most decent way to solve this problem is looking to the methods evolved by human composers over 1000 years.(Real solution)
- Again, for solving this problem, we will need to enhance the very promissing constraint-based approaches with explicit meta-level reasoning controlling search to render this enormous solution space tractable.

Using Constraints for Serial Composition – Serial Composition

- Schoenberg(1984)
- A richly chromatic Romantic composer
- Chromaticism
- From late Romantic period, Richard Straub and Wagner, more and more notes were added in to the composers palettebeyond the basic seven of the major and minor scales predominantly used by the Classical composers.

Arnold Schoenberg, Los Angeles, 1948

Background information

Birth name Arnold Schoenberg

BornSeptember 13, 1874

OriginLeopoldstadt, Austria

DiedJuly 13, 1951,United States

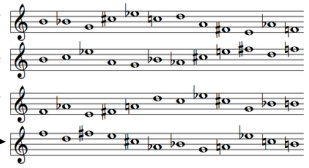
Occupation(s) Composer

Twelve note method

- Composing a piece using the twelve notes of the chromatic scale in strictly equal numbers, by building a series-a twelve note chromatic sequence in which no note is repeated
- Mathematically, every note have own frequency, for example A4 = 440Hz, E5 = 660Hz, C# = 550Hz
- Why this important in twelve note music?
  - Schoenberg justified his approach of using the chromatic, or twelve-note scale, rather than the conventional diatonic one, which gives dominance to the lower harmonic ratios, on the basis that it was simply an extension of that system to the higher harmonic ratios.
  - Closer notes to the unity ratio from the original notes tends to make a unpleasant sound perceptually. “Clash” or “Dischord”
  - Schoenberg said simply stated that “dissonance is an outmoded concept.”

Serial Composition(Twelve note composition)

- Four basic operations
  - Prime
  - Inversion
  - Retrograde
  - Retrograde inversion



Furthermore

- Appearances of P can be transformed from the original in three basic ways:
- transposition up or down, giving Pχ.
- reversal in time, giving the *retrograde* (R)
- reversal in pitch, giving the *inversion* (I): I(χ) = 12 - Pχ.
- The various transformations can be combined. The combination of the retrograde and inversion transformations is known as the *retrograde inversion* (*RI*).
- RI is:RI of P,R of I,and I of R.R is:R of P,RI of I,and I of RI.I is:I of P,RI of R,and R of RI.P is:R of R,I of I,and RI of RI.

Case Study : Generating a Series 1. Task Description

- 12! possible cases
- “My Favorite chord, based on A”
- Series to begin with the note “G”

- A new piece for flute, oboe, ’ cello and harp, called “Elements”
  - Flute : the original series, played linearly;
  - Oboe : the retrograde series;
  - ‘Cello : the original series, rotated four places;
  - Harp : the retrograde series, rotated eight places;
- These constraints are easy to implement under “Constraint Logic Programming.”

Case Study : Generating a Series 2. Approach(1)

- Constraint Logic Programming over Finite Integer Domains that a very natural way to represent a series is with a list of integer variables constrained to lie between 0~11.
- Implementation Series Constrained

```
series( Fl ) :-
        is_row( Fl ),              % Fl is Flute part
        retrograde( Fl, Ob ),      % Ob is oboe part
        rotate( 4, Fl, Vc ),       % Vc is ’cello part
        rotate( 8, Ob, Hp ),       % Hp is harp part
        chords( Fl, Ob, Vc, Hp ).
```

Case Study : Generating a Series 2. Approach(2)

- is_series(Series) :- function
- retrograde(Forwards, Backwards) :- function
- rotate(N, Initial Rotated) :-

```
chords( [A|As], [B|Bs], [C|Cs], [D|Ds] ) :-
        domain( [P,Q,R,S], 0, 11 ), % make a store for the sorted list
        ( P + 3 ) mod 12 #= Q ,     % apply intervals needed to create
        ( Q + 1 ) mod 12 #= R,      % the chord, mod 12 to prevent
        ( R + 3 ) mod 12 #= S,      % overshooting the octave
        all_different( [E,F,G,H] ), % each note must be picked once
        element( E, [A,B,C,D], P ), % pick notes until they are sorted
        element( F, [A,B,C,D], Q ),
        element( G, [A,B,C,D], R ),
        element( H, [A,B,C,D], S ),
        chords( As, Bs, Cs, Ds ).   % and recurse along the series
```

Case Study : Generating a Series 3. The Series & Discussion

- Many solutions to the constraint system shown above
- Choose Figure 7 as a symmetrical series constrained around "False-relation" chord

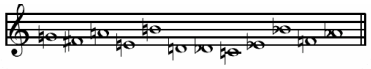


Figure 7: Symmetrical series constrained around "false-relation" chord

- Empirical study of how constraint logic programming can be useful in music composition.
- Integer finite domains

Musical Representation and Variable Temperament

- Pitch system, called temperament
- Two temperament
  - Just Temperament
    - Human pitch perception
    - Real Instrument based on harmonic series of a tube

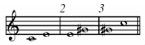


Figure 8: Three major thirds, making an octave

$$\frac{5}{4} \times \frac{5}{4} \times \frac{5}{4} = 1.953$$

  - Equal Temperament
    - Mathematical systems

$$\sqrt[12]{2} \simeq 1.059$$

Musical Representation and Variable Temperament - Discussion

- Those two temperament constraint system required in the two different respects, that is to say, both in mathematical or physically accurate system and in human acoustics world

Conclusion

- This article is for those who would use constraint technology in the creation of music and the simulation of human musical behaviour.
- Firstly, four-part harmonisation ; search space is not amenable to unstructured searching, even with the power of constraint technology, due to its intensely convoluted nature.
- Secondly, he outlined a simple system designed to help himself in the composition of equal-tempered, twelve-note composition, and suggested that Constraint Programming with Integer Finite Domain is a very good solution.
- Lastly, there are two different temperament, equal-temperament and just-temperament. First one is good for tonal music generation but still need second one, because real human's singing and tube based acoustics instrument such as pipe organ, xylophone and almost every brass(trumpet, trombone, horn, tuba) and woodwinds(clarinet, piccolo, flute, oboe, etc).