Journal of Mathematics and Music, **INSERT YEAR**
Vol. **INSERT**, No. **INSERT**, INSERT-DOI-LINK-FROM-PUBLISHER

Online Supplement to "Data-based melody generation through multi-objective evolutionary computation"

This Online Supplement contains musical examples for the above mentioned paper. These monophonic melodies have been generated using a graphical user interface prototype running an evolutionary computation system that generates melodies of eight bars that are evaluated by a multi objective fitness function trained on a target melody dataset.

Keywords: melody; evolutionary computation; multi objective evaluation; genre; style

2010 Mathematics Subject Classification: 68T05
2012 Computing Classification Scheme: Applied computing; Sound and music computing; Computing methodologies; Learning from critiques; Computing methodologies; Genetic programming;

1. Introduction

Genotype structure

The melodies presented here are represented in that system by trees derived from the following grammar:

```
Melody := Section Section
Section := Measure Measure Measure Measure
Measure := (Beat | Symbol) (Beat | Symbol) (Beat | Symbol) (Beat | Symbol)
Beat := (SubBeat | Symbol) (SubBeat | Symbol)
SubBeat := Symbol Symbol
Symbol := Pitch | Rest | Continuation
Pitch := C | D | E | F | G | A | B
Rest := r
Continuation := _
```

Evolutionary computation configuration

All twelve evaluator functions presented in the paper were used as combined objective functions. The NSGA-II multi objective fitness function was used to rank the melodies breed in each generation. Each melody has been drawn from a population of one hundred individuals evolved over one hundred generations, using crossover, mutation, and reproduction in the following proportions:

crossover: 90%
mutation: 5%
reproduction: 5%

Data sets

In total, nine different sets of MIDI files, containing one monophonic melody track per file, corresponding to nine different genres of music have been used to train the fitness evaluators. Music files from three 'domains' have been utilized: popular, jazz, and academic music. Popular music data have been separated into three subgenres: pop, blues, and celtic (mainly Irish jigs and reels). For jazz, three styles have been established: a pre-bop class grouping swing, early, and Broadway tunes, bop standards, and bossanovas as a representative of latin jazz. Finally, academic music has been categorized according to historic periods: baroque, classicism, and romanticism.

The number of files eventually used for each genre is displayed in Table 1. The total amount of pieces was 856, providing 47 and a half hours of music data.

Academic	235	Jazz	338	Popular	283
Baroque	56	Pre-bop	178	Blues	84
Classical	50	Bop	94	Pop	100
Romanticism	129	Bossanova	66	Celtic	99

Table 1.: Number of files per genre and subgenre in melody dataset.

A different run of the evolutionary computation system has been executed for each of the nine datasets. From each run, two representative melodies are shown below:

2. Generated melodies

Academic/baroque

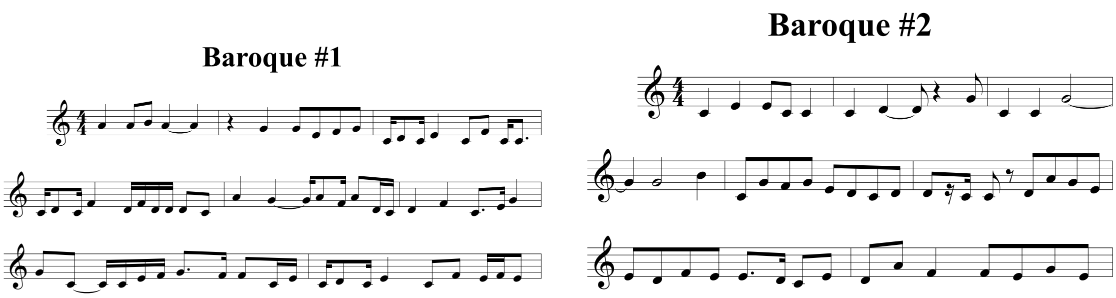

Figure 1.: Baroque melodies.

Academic/classicism

Figure 2.: Classicism melodies.

Academic/romanticism

Figure 3.: Romanticism melodies.

Jazz/bop

Figure 4.: Be-bop melodies.

Jazz/bossanova

Figure 5.: Bossa-nova melodies.

Jazz/pre

Figure 6.: Pre-bop jazz melodies.

Popular/blues

Figure 7.: Blues melodies.

Popular/celtic

Figure 8.: Celtic melodies.

Popular/pop

Figure 9.: Pop melodies.

3. User interface

A snapshot of the user interface utilized to generate the above melodic compositions is shown in figure 10. The selectable set of fitness evaluators is on top of the interface window. Some features of the interface not described in the paper are listed below:

- Several key profiles can be selected, including major and minor scales, or Krumhansl-Kessler profiles (Krumhansl and Kessler 1982).
- The system can be trained on any folder containing a set of MIDI files encoding melodies as monophonic tracks.
- A configuration file is used to setup the evolutionary computation system.
- Some configuration parameters can be overriden from the interface, like the number of generations, or the population size, for example.
- A seed controls the evolutionary computation system inicialization. Given a fixed configuration, the same seed produces exactly the same result.
- The generated melodies can be exported to the MusicXML format (Selfridge-Field 1997). The genotype can be represented as a tree in DOT format (Graphviz and format 2016).
- The actual melody fenotype can be visualized in western music notation, and played to the default audio output device.

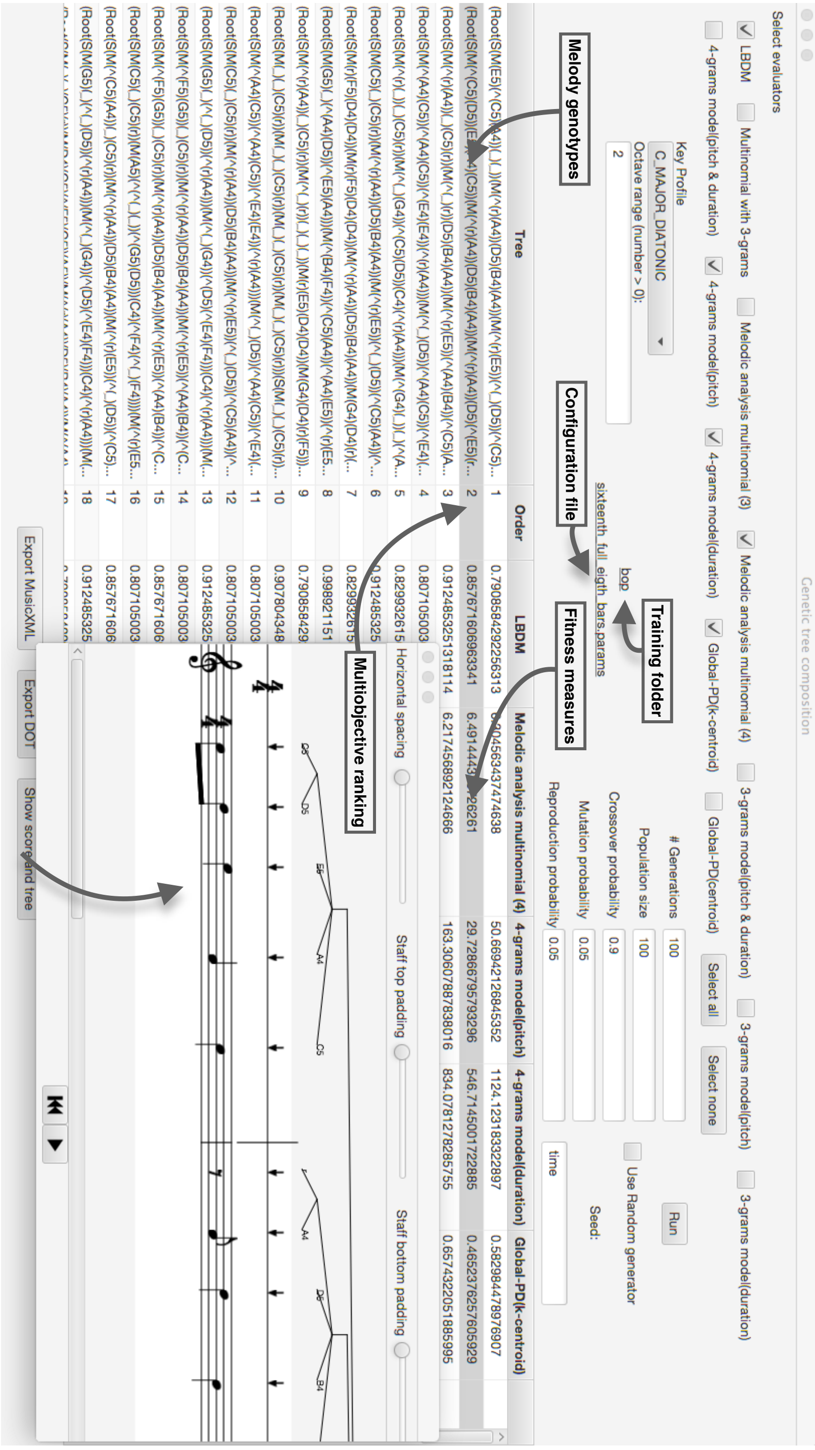

Figure 10.: User interface snapshot

4. Combined fitness evolution

Figures 11, 12, and 13 show the evolution over generations of the Pareto optimal frontier for several pairs of fitness functions. All runs have been executed with the same configuration: a fixed seed, a major scale profile over two octaves, breeding a population of one hundred individuals over one hundred generations with a crossover probability of 90%, and mutation and reproduction probabilities of 5%. Each figure shows a run using a fitness function from one of the three fitness families against a function from the other two families.

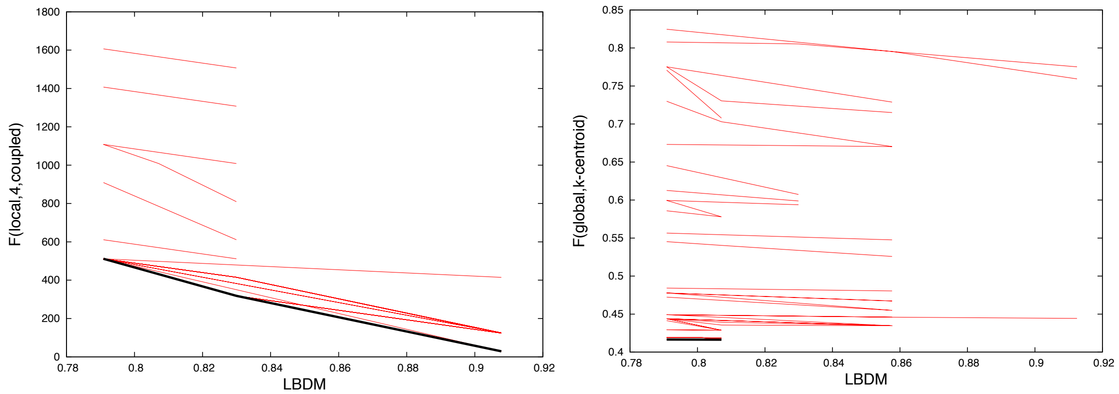

Figure 11.: (left) LBDM vs. $F_{\text{local},4}$ using coupled pitch and duration pareto frontiers. While the system is capable of minimizing $F_{\text{local},4}$ over generations, it seems unable to improve the LBDM fitness of the population at the same time. (right) LBDM vs. F_{global,d_k} pareto frontiers. Last generation degenerates to a single point in the frontier.

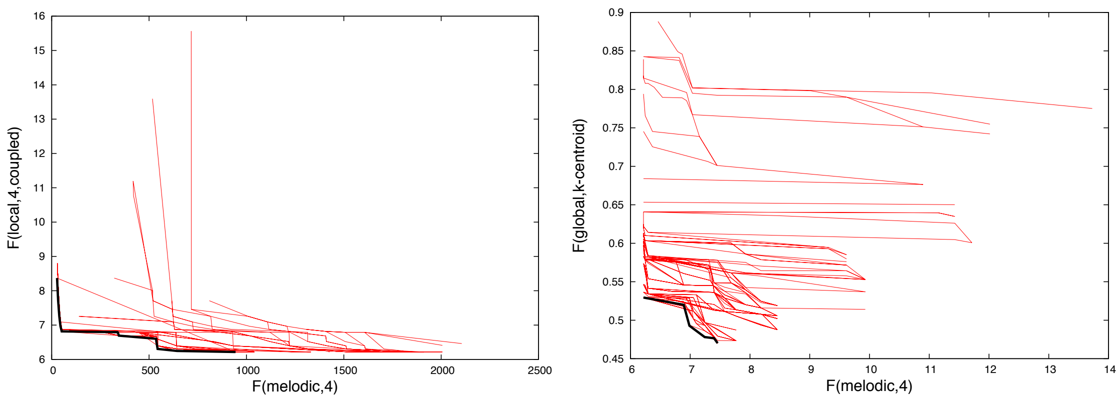

Figure 12.: (left) Pareto frontiers over generations for $F_{\text{melodic},4}$ vs. $F_{\text{local},4}$ using coupled pitch and duration. (right) Pareto frontiers over generations for $F_{\text{melodic},4}$ vs. F_{global,d_k}. The evolution of the fitness in both figures shows that the system can find a way to improve both fitness measures at the same time.

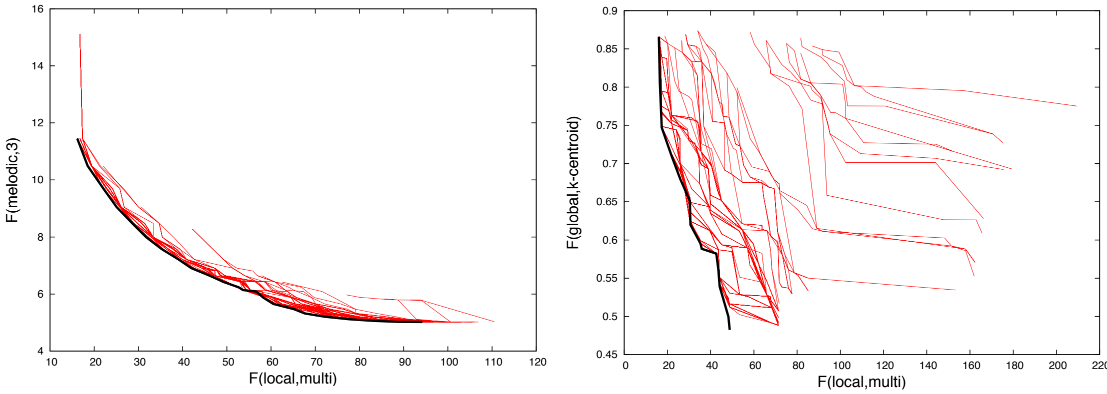

Figure 13.: (left) Pareto frontiers over generations for $F_{\text{local},multi}$ vs. $F_{\text{melodic},3}$ using coupled pitch and duration. (right) Pareto frontiers over generations for $F_{\text{local},multi}$ vs $F_{\text{melodic},3}$. The evolution of the Pareto frontier, specially in the left figure, shows that these measures work well when used together, as the system can find a way to improve both of them simultaneously, albeit somewhat slow in the left run.

References

Graphviz, and DOT format. 2016. "http://www.graphviz.org/Documentation.php." Accessed 20160222.

Krumhansl, Carol L, and Edward J Kessler. 1982. "Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys.." *Psychological review* 89 (4): 334.

Selfridge-Field, Eleanor. 1997. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press.