An evolutionary algorithm for harmonic music composition

José Pedro Aguerre, Rodrigo Bayá,
Renzo Massobrio, and Sergio Nesmachnow

Universidad de la República, Uruguay

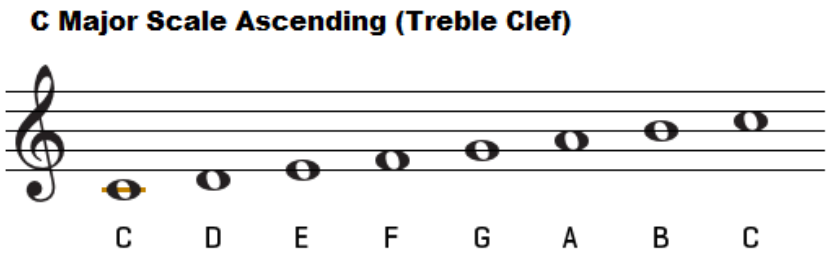
Metaheuristics International Conference, Barcelona, 2017

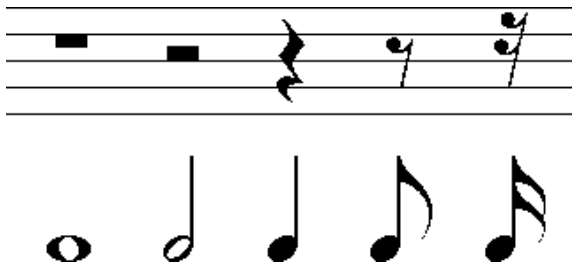
UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Outline

The harmonic music composition problem

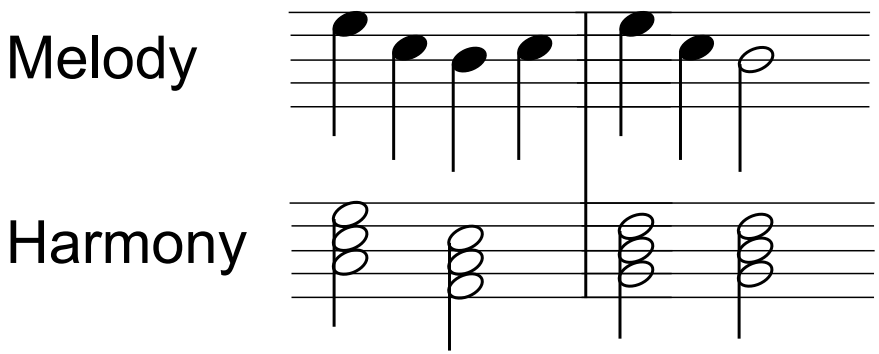
- Music can be represented as scores.
- A score consists of rests and notes with their octave and duration.
- Notes are C, D, E, F, G, A, B, and their respective sharps. Duration can be whole, half, quarter, etc.
- Other concepts, such as compasses and key and time signatures, are not considered in the problem formulation.





The harmonic music composition problem

- Harmonic music composition is the process of defining a set of notes representing melodies and harmonies.
- A melody is defined as an ordered set of notes.
- Harmony is the union of two or more melodies that generate harmonic intervals.



The harmonic music composition problem

- Mathematical formulation

  Find a score s_G, given:

  - A set of reference scores $S = \{s_1, \ldots s_N\}$, $|S| = N$;
  - A function $f : \{s_G\} \times S \rightarrow \mathbb{N}$ that evaluates similarity between two scores.

  The goal of the problem is to maximize the objective function OF, where $\alpha, \beta \in [0, 1]$ and $\alpha + \beta = 1$.

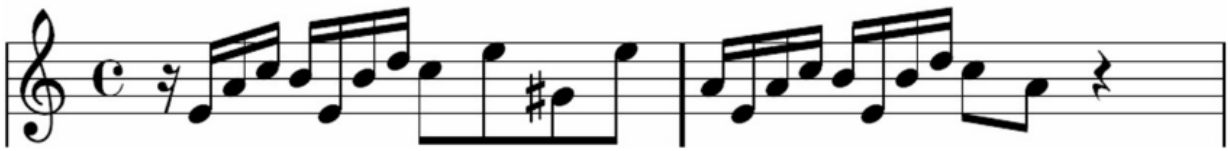
$$OF = \max \left\{ \alpha \times \underbrace{\frac{\sum_{s \in S} f(s_G, s)}{N}}_{\text{average similarity}} + \beta \times \underbrace{\max_{s \in S} f(s_G, s)}_{\text{closest score}} \right\}$$

Related work

- Studies on automatic music generation date back to the late 1950s.
- Several proposals, including: random numbers, formal grammars, cellular automata, fractals, and evolutionary computation.
- A summary of methods on automatic music composition is presented in Miranda (2001).
- Three main approaches:
  - generate completely random music.
  - construct compositions based on music rules, considering either melodies or harmonies.
  - generate compositions using entry sounds or scores.
- No previous attempts using EA to exploit the similarity between compositions by the same author.

EA for harmonic music composition

- We propose a parallel evolutionary algorithm, following the distributed subpopulations model.
- Solution encoding:
  - A musical score is represented by a string.
  - A note is represented by a set of characters: i) the note ii) "#" if the note is sharp; iii) the octave iv) the duration: *w*–whole, *h*–half, *q*–quartet, *i*–eighth and *s*–sixteenth.
  - *R* represents a rest, followed by its duration.
  - *V* represents a new voice.

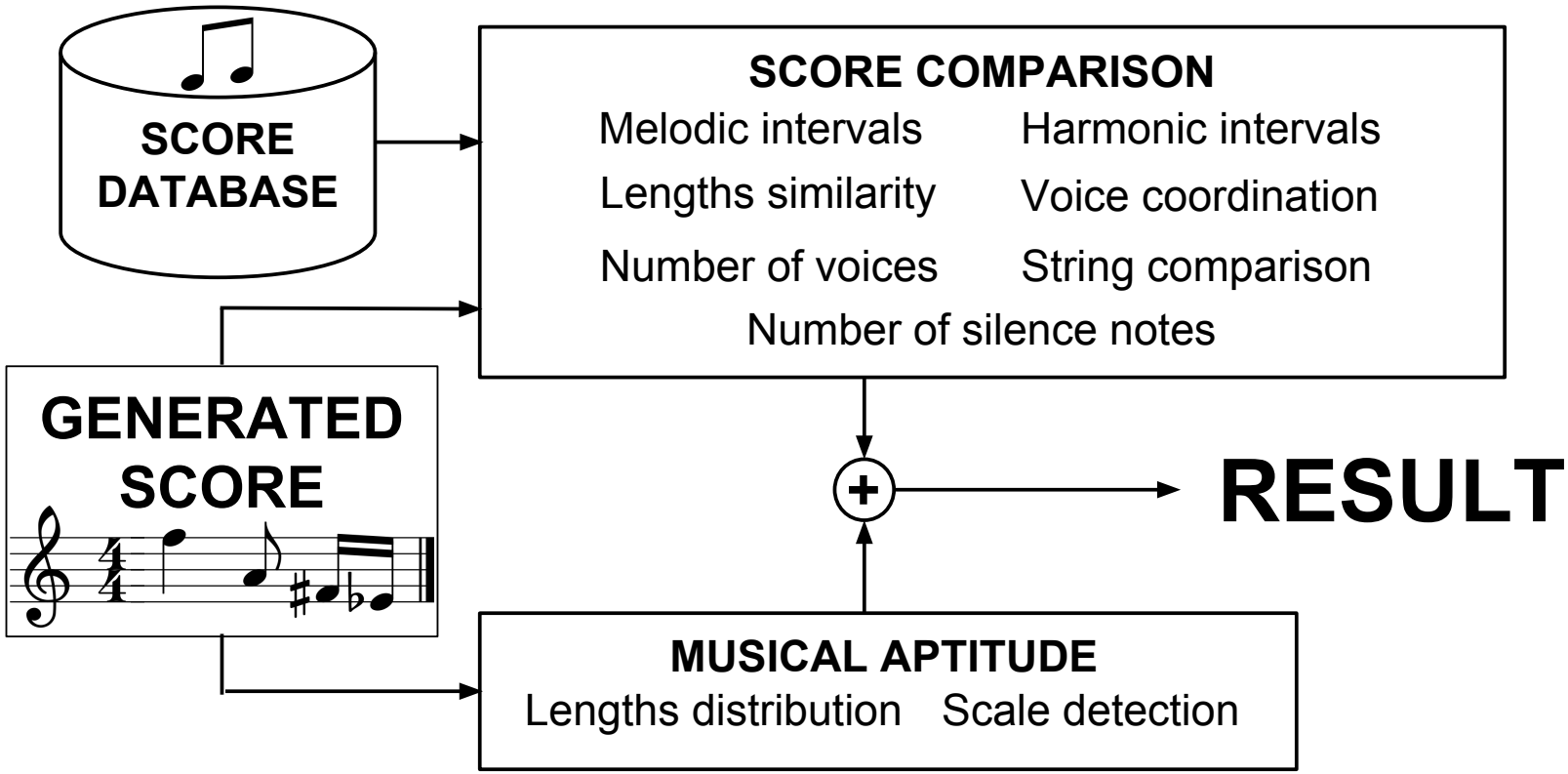


"Rs E5s A5s C6s B5s E5s B5s D6s C6i E6i G#5i
E6i A5s E5s A5s C6s B5s E5s B5s D6s C6i A5i Rq"

Example: extract of Bach's Inventio 13

EA for harmonic music composition

- Fitness function: seven score comparison criteria and two musical aptitude metrics.



EA for harmonic music composition

- Evolutionary operators:
  - *Selection.* Stochastic universal sampling operator.
  - *Recombination.* Single point list crossover.
  - *Mutation.* Four operators:
    1. random note modification.
    2. add or delete note.
    3. split or join note.
    4. voice order permutation.
  - *Migration.* Periodically exchanges randomly selected individuals between demes, considering an unidirectional ring topology.
- Evolution strategy: generational gap of populations, with elitism.
- Stopping criterion: stagnation during 10 epochs.

Results and discussion

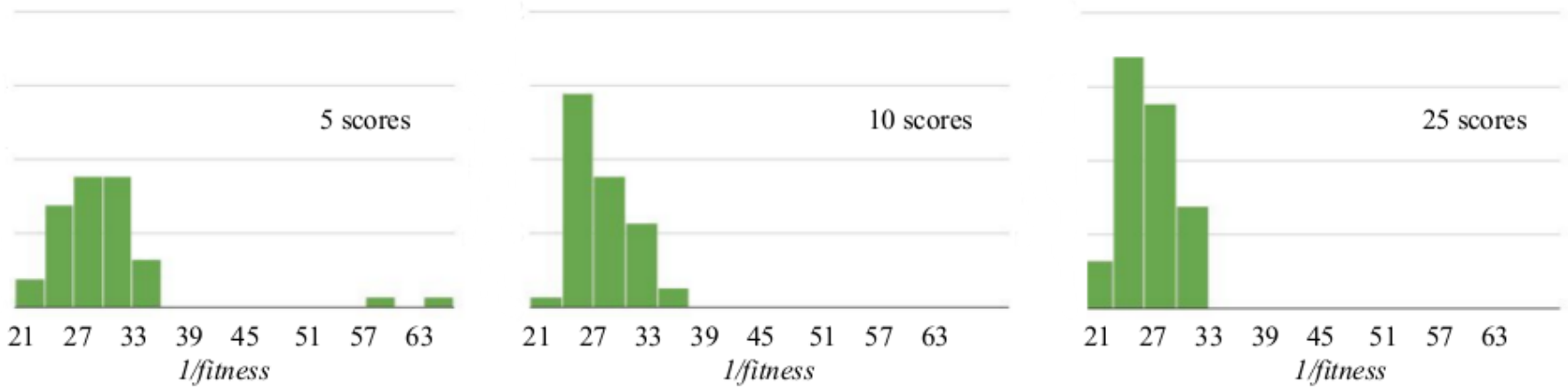
- Software and hardware platform:
  - EA implemented in Java using the Watchmaker framework.
  - Executions performed on a core i7-3770, 16 GB RAM, and Fedora 19.
- Parameter configuration experiments.
  - Preliminary experiments:
    - ★ Weights: $\alpha = 0.3, \beta = 0.7$.
    - ★ Number of demes: 8.
    - ★ Recombination probability: 0.75.
    - ★ Elite population: 5%.
  - 20 independent executions on a set of 10 scores by Bach and a stopping criterion of 400 generations.
    - ★ Mutation probability: $\{0.001, 0.005\}$.
    - ★ Generations per epoch: $\{40, 50\}$.
    - ★ Population per deme: $\{80, 100, 120\}$.
    - ★ Migrants per epoch: $\{5, 10\}$.

Results and discussion

- 3 databases with 5, 10, and 25 scores by Bach.
- 50 independent executions for each instance.

# scores	*1/fitness* ($\mu \pm \sigma$)	*time (s)* ($\mu \pm \sigma$)
5	30.1±2.6	454.8±113.4
10	27.3±2.6	413.6±108.9
25	27.1±2.9	378.9±95.5

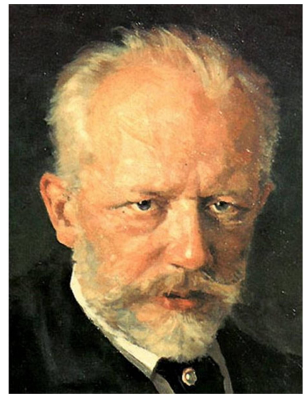
Better musical properties achieved with larger reference databases.



Results and discussion

- Similar results when using other composers as reference, on classical and rock music.






	Mozart	*Beethoven*	*Chaikovsky*	*The Beatles*
1/fitness ($\mu \pm \sigma$)	31.5±2.6	28.5±2.0	32.1±2.7	30.1±2.5
time (s) ($\mu \pm \sigma$)	414.2±101.1	393.5±99.3	418.31±109.7	390.67±96.4

Conclusions and future work

Conclusions:

- A parallel EA for harmonic music composition was presented.
- Fitness function uses score comparison against a reference database and musical aptitude metrics.
- Better musical attributes are obtained with larger reference databases.
- Accurate results are reported for three different Bach databases and for other classic and rock composers.

Future work:

- Include other musical aptitude criteria.
- Execute in cluster/cloud to deal with larger databases.
- Evaluate other metaheuristics for harmonic music composition.

Conclusions and future work

Conclusions:

- A parallel EA for harmonic music composition was presented.
- Fitness function uses score comparison against a reference database and musical aptitude metrics.
- Better musical attributes are obtained with larger reference databases.
- Accurate results are reported for three different Bach databases and for other classic and rock composers.

Future work:

- Include other musical aptitude criteria.
- Execute in cluster/cloud to deal with larger databases.
- Evaluate other metaheuristics for harmonic music composition.

Thanks for your attention