Populations of Populations: Composing with Multiple Evolutionary Algorithms

Arne Eigenfeldt[1], Philippe Pasquier[2],

[1] School for the Contemporary Arts, Simon Fraser University
Vancouver, Canada
[2] School for Interactive Arts and Technology, Simon Fraser University,
Surrey, Canada
{arne_e, pasquier)@sfu.ca

Abstract. We present a music composition system in which musical motives are treated as individuals within a population, and that the audible evolution of populations over time are of musical interest. The system additionally uses genetic algorithms to generate high level musical aspects that control how the population is presented, and how it may be combined with other populations. These algorithms feature fitness functions that adapt based upon context: specifically, by using an analysis of the evolving population, the fitness functions adjust their constituent parameters in selecting strong individuals.

Keywords: Biologically inspired music; genetic algorithms, evolutionary music.

1 Introduction

Evolutionary algorithms have been successfully used in music composition [1-3], [10], [15-16] for many reasons. One aspect relates to the notion of musical development – the evolution of musical ideas over time – and its relationship to biological evolution. As music is a time-based art, the presentation of successive generations – rather than only the final generation – allows for the aural exposition of evolving musical ideas. The concept of organic development has been a paradigm within music composition for centuries [4], and continues to be so in contemporary music [7]. We present a music composition system in which musical motives are treated as individuals within a population, and that the audible evolution of populations over time are of musical interest. The system additionally uses genetic algorithms to generate macro-level aspects that control how a population is presented, and how it may be combined with other populations. These algorithms feature fitness functions that adapt based upon context: specifically, by using an analysis of the evolving population, the fitness functions adjust their constituent parameters in selecting strong individuals. This system has been used to generate a set of pieces for solo percussionist, as well as a work for marimba, violin, and piano.

1.1 Goals

This system was created with the sole intention of generating "art-music": music for the concert stage. While the concept of an evolving population of individuals that consist of rhythmic figures was seminal from the outset, the use of genetic algorithms to determine high level compositional decisions was a later addition, but one arrived at for purely musical reasons.

Waschka gives a revealing description of the problem faced by composers of contemporary concert music, their relation to their material, and the notion of the 'well-defined problem space'[15]. To paraphrase, he points out that the desired solution of "good new music" is not, in itself, clearly defined: "Most composers, upon hearing a piece, even for the first time, feel confident of their ability to judge its quality and believe they will be able to point out what things about the piece worked well and what did not. However, such estimations differ significantly from knowing, a priori, what will make a good, non-formulaic, experimental, or avant-garde piece" [15]. This points directly to the inherent complication of using evolutionary algorithms within music: the difficulty in designing a non-interactive fitness function. Waschka solves this problem by avoiding the issue entirely, and selecting individuals for reproduction through random methods; others [2, 6], have lessened the burden of separating strong and weak individuals by initializing the population with what are already determined to be strong individuals. We have chosen to approach the problem in a similar fashion: no fitness function was used in evolving the population of rhythmic motives – it was assumed that each generation was comprising of interesting (i.e. strong) individual elements. However, as described in Sections 3.3 and 3.4, fitness functions were used in how the generations were selected for presentation.

1.2 Overview

Our system begins with a relatively small population (between four to twelve individuals) that is generated by the user through probabilistic methods over various parameters (see Section 3.1). Initially, each individual represents one beat; however, during evolution, individuals can combine to form longer units (see Section 3.2). Throughout evolution, the population remains ordered. The temporal sequence of all the individuals in the population stands for a musical phrase.

The user can adjust the probability of how operators evolve the population, both individually or collectively: these probabilities can change over successive generations (see Section 3.2). Once a series of generations have been created – which can be considered the history of the population – these populations are analysed to determine variation over the generations (see Section 3.3). This analysis is then used to create a trajectory through these generations, through the use of a genetic algorithm whose fitness functions vary depending upon the analysis of the population (see Section 3.3). The trajectory is then used to select the order and repetition of generations over time, which results in the succession of musical phrases within a section of music. As it is often desirable to present more than a single musical idea at a time, the system alternates between various trajectories of different population histories, a process we refer to as braiding (see Section 3.4).

1.3 Musical Considerations

The design of the system was engendered by specific musical goals that the first author – a composer of concert music – considered "interesting" through an auto-ethnographic analysis of his own music. They are, in no particular order:

- *Repetition*. Musical phrases may be repeated directly before being varied; however, repetition is not the focus of the music. This can be seen as emblematic of "post-minimalism", in which the pure repetitive structures of minimalism are forgone, while a degree of repetition remains [8]. In evolutionary terms, a specific population may be presented more than once, in succession.
- *Additive processes*. Musical phrases need not remain a consistent length, and the addition and subtraction of beats is an important element of variation. In evolutionary terms, the population is not constrained to a consistent length.
- *Processes that are not sequential*. The amount variation between phrases is not constant, and can include the presentation of phrases already heard. This is similar to the additive processes employed by Philip Glass: "In Glass's music, linear additive process is somewhat more flexible: only rarely in his works do the melodic units grow by the addition of only one note at a time" [14]. In evolutionary terms, this results through the presentation of non-successive generations, as well as the possibility of presenting generations already heard.
- *Block additive process*. The potential for unfolding a musical phrase of a set length through the replacing of rests by beats is possible. This is a standard technique employed by Steve Reich, which "consists of the gradual assembly of a unit within a predetermined and unchanging time frame" [14]. In evolutionary terms, this results through the replacement of null individuals with those containing musical representations.
- *Developing variation*. Since at least the time of Bach, concepts of development and variation have been coupled, since variation is produced through the development of existing material. The term "developing variations" was suggested by Arnold Schoenberg, who considered it to be one of the most important compositional principles since 1750: "variation of the features of a basic unit produces all the thematic formulations which provide for fluency, contrasts, variety, logic and unity, on the one hand, and character, mood, expression, and every needed differentiation, on the other hand - thus elaborating the idea of the piece" [9]. In evolutionary terms, this results from operating upon successive generations, rather than limiting the evolution to a single generation (with many individuals).
- *Splicing technique*. Musically, this contrasts developing variation, in that the musical flow is suddenly interrupted by divergent material. Notably used by

Stravinsky [5], it became a fundamental compositional tool of minimalist and post-minimalist composers, who juxtaposed several musical processes set in motion within a composition: "systems music involves not one but a number of such processes. These do not necessarily occur simultaneously...one process may abruptly switch to another, as if two independent pieces had been cut up and spliced together" [14].

Section 2 will discuss related work; Section 3 will present a detailed description of the system; Section 4 will offer a conclusion and future research.

2. RELATED WORK

Evolutionary computation has been used within music for over two decades in various ways. Todd and Werner [13] provide a good overview of the earlier musical explorations using such approaches, while Miranda and Biles [11] provide a more recent survey. Very few of these approaches have been compositional in nature – using evolutionary methods to generate entire compositions rather than portions of compositions; instead, their foci is upon generating melodies, harmonies, or timbre.

Several real-time applications of GAs have been used, including [16], which selected individuals from an Interactive Genetic Algorithm (IGA) suitable for the immediate situation within a real-time improvisation. Another approach was by Beyls [1] in which the fitness function sought either similar individuals or contrasting individuals to an immediate situation within an improvisation.

Thywissen [12] describes a system that allows composers to evolve musical structures interactively. Of note is the consideration of higher-level musical structures, which he calls meta-compositional grammars.

Waschka [15] used a GA to generate contemporary art-music music, which more closely resembles the goal of our system. His explanation of the relationship of time within music is fundamental to understanding the potential for evolutionary algorithms within art-music: "unlike material objects, including some works of art, music is time-based. The changes heard in a piece over its duration and how those changes are handled can be the most important aspect of a work." Waschka's *GenDash* has several important attributes, several of which are unusual: an individual is a measure of music; all individuals in all generations are performed; the fitness function is random, leading to random selection; the composer chooses the initial population. Of note is the second stated attribute, the result of which is that "the evolutionary process itself, not the result of a particular number of iterations, constituted the music". Waschka provides some heuristic justifications for his choices, suggesting that while they may not make sense in the natural world, they do provide musically useful results.

3. SYSTEM DESCRIPTION

Fig. 1 gives an overview of the system.

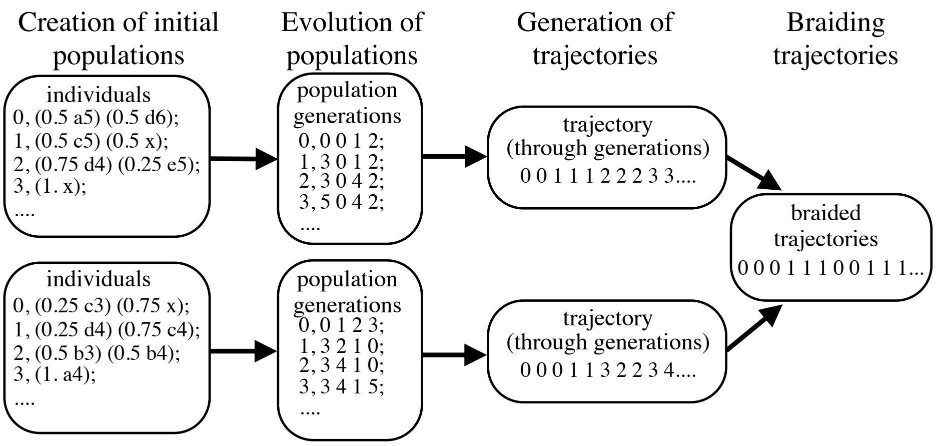

Fig. 1 Individuals are generated by the user (see Section 3.1) and populations are evolved (see Section 3.2). A genetic algorithm is used to generate a trajectory through the population generations (see Section 3.3). Trajectories can be combined through braiding to form larger musical sections (see Section3.4). Braided trajectories form a composition.

3.1 Creation of the Initial Population

The creation of the initial population is a crucial stage in determining the contents of successive generations. Rather than requiring the user to pre-select individuals for the initial population, generative methods have been employed that are more consistent with the principles underlying the system. The user selects the approximate size of the initial population by determining a range (from 2 to 10 individuals), as well as the maximum number of generations (which corresponds to the number of unique phrases within the resulting composition). As these generations are later braided with another population's generations, coupled with the possibility of phrases being repeated, there has never been a need to exceed 50 generations.

The user determines the metric value for all individuals within the population, either an eighth-note, quarter-note, or half-note. The number of onsets within an individual determines its density, which, when considered over the population, is an important defining feature of the population as a whole. The change in density over the individuals within the ordered population correlates to the change in activity over the course of the musical phrase. For this reason, the user can indicate the overall density of the initial population, as well as how that density varies over the population.

An individual's density is correlated to the possible number of subdivisions of its overall duration. As such, the user can adjust the relative probabilities for various subdivisions (see Fig. 2). Each individual is comprised of a single subdivision, ranging from 1 (the user set metric value, such as a quarter note) to 10 (sixteenth-note quintuplets, in the case of a metric value of a quarter note).

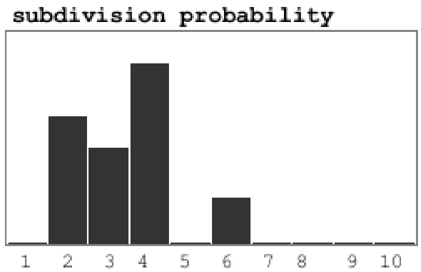

Fig. 2 Probability distribution for subdivision of the user-set metric value within an individual. For example, given a metric value of a quarter note, a subdivision of 3 results in a triplet.

Pitch probability is determined in a similar manner: as the initial musical output of the system was for solo percussionist, only a limited number (up to eight) of fixed pitches were possible.

The user also determines the likelihood of double onsets occurring for any onset, which translates into two notes being struck at the same time. Lastly, the user determines coefficients for metric and rhythmic consistency between individuals; during generation, the previously selected subdivision and/or pitch may be "held over" if the user selects a high consistency.

An individuals genotype is represented as [[onset time within beat] & [pitch-name + octave]] (see Fig. 3): the onset time is a percentile of the relative duration of a beat.

0, [0.25 x] [0.5 f6] [0.25 e5];
1, [0.166667 b6] [0.166667 b6] [0.666667 f6];
2, [0.4 e5] [0.6 e5];
3, [0.4 x] [0.6 f6];
4, [0.5 f6] [0.5 f6];

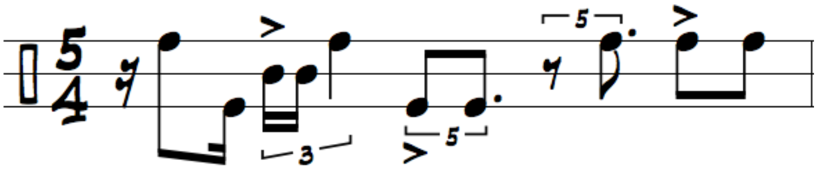

Fig. 3 The representation for the first five individuals as text, and as musical notation. Pitch x indicates a rest.

As individuals are generated, they are compared to the list of existing individuals: if the individual does not exist, it is added to the database, and its index recorded within the population list; if the individual does exist within the population, that individual's index is used.

Populations consist of indices to the Individual array; thus, the population for Fig. 3 would be (0 1 2 3 4), as they are the first individuals used, and no repetition occurs.

A separate array is used to store accents, which are generated independently, and are not discussed here for reasons of brevity.

3.2 Evolution of Populations

Populations evolve through both individual and population operators, over a number of generations set by the user. Population operators alter the order of individuals, but not the individuals themselves. These include: adding new or replacing existing individuals (both of which trigger the Generate algorithm for a single individual); dropping individuals; shuffling individuals; duplicating individuals. Individual operators alter the individuals themselves. These include: changing a rest within the individual into an onset; changing an onset into a rest; altering a pitch; changing an onset. More than one operator can affect the population per generation, the maximum number of operations per generation is set by the user.

The user can set the probability weight for all operators, which can change over time through dynamic function generators (see Fig. 4). For example, the user could set only Shuffle and Replace as population operators for the first half of all generations (which would maintain the same population size over those generations), and then increase the probabilities for Add, Drop, and Copy for the second half of the generations (which would alter the population size).

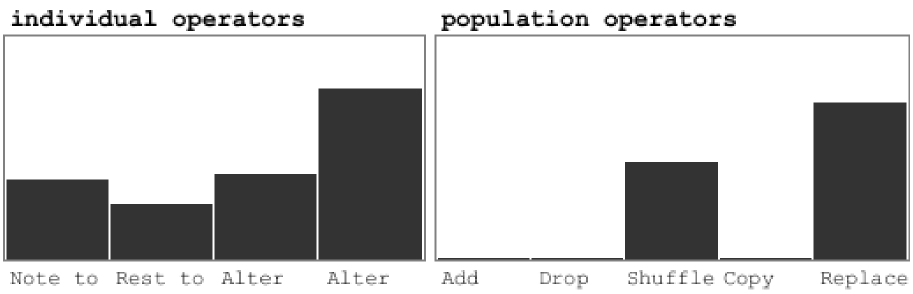

Fig. 4 Probabilities for Individual and Population operators for a generation.

As the generations are being evolved, an algorithm keeps track of which individuals have remained adjacent to one another; in other words, which individuals will eventually be heard as larger groups over time. Once the number of generations has passed – set by the user – the individuals are concatenated into a new, larger individual. In subsequent generations, this new group will be treated as a single individual by the operators: thus, it can be copied, shuffled, dropped, or altered in its entirety.

3.3 Analysis of Populations and Trajectories

When the first iteration of the system was completed, and the musical results were viewed, the sequential progression from the initial population through its evolved histories presented a clear development of material – computational evolution directly equated to musical evolution. However, such a simple and predictable sequential progression was judged to be artistically limited, and, for musical reasons, a method was needed to negotiate through the generations that included not only the repetition of selected phrase populations, but also a progression that allowed for non-sequential selection, as well as the recurrence of older generations. A random walk through the generations was considered unsatisfying, in that selections were made for unmusical reasons; instead, what was required was some consideration of the contents of the populations themselves. When manually selecting which populations should repeat, be skipped, and returned to, auto-ethnographic analysis revealed certain predilections that were desired within a selection algorithm, which became the algorithm's heuristics:

- a tendency for sequential motion – beginning from the first generation and progressing, more or less, toward the last;
- repetition of generations; however, those generations that are considered "less interesting" should not be repeated as often as those that are considered "more interesting";
- a mixture of contiguous and non-contiguous generations, with the occasional large deviation;
- as many of the generations as possible should be included.

An intermediary stage between generation and audition was therefore inserted, in which each generation and its individuals were analysed. This analysis determined:

- the population's overall density (number of onsets / population size);
- density variation between the population's individuals;
- mean rhythmic complexity of the population (the degree of syncopation within the individuals and their subdivision);
- complexity variation between the population's individuals;
- mean similarity of the population's rhythms;
- similarity variation between the population's individuals.

The analysis of all the generations of a phrase population was then used by a genetic algorithm (GA) to determine the best trajectory through the generations. In this GA, a population of 100 individuals is generated, the length of an individual corresponding to the number of total phrases requested by the user. The individuals consist of step sizes, where 0 represents no change (repetition of a phrase), positive values represent a forward progression through the generations, and negative values represent a backward progression. The constraints of the selection algorithm essentially produced variations of the shapes shown in Fig. 5. For this reason, the initial population for this stage was not random, but variations of this shape, generated using variable parameters set by the user, including maximum step size, step size variation, direction, and direction variation (see Fig. 5). Due to this non-random initialisation, it was found that only five generations were required in order to successfully evolve.

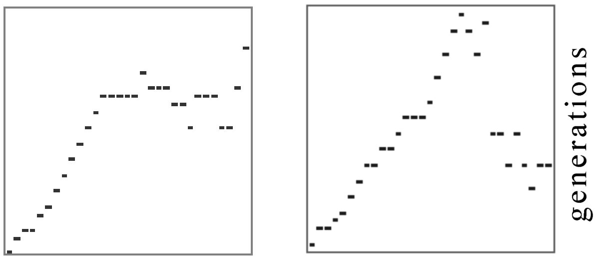

Fig. 5 Two trajectories through phrase population generations: left, initial generation; right, after five generations of evolution. Horizontal axes indicate time, vertical axes indicate generation number. While the shapes are similar, the specific values that are selected in the second trajectory, including those that are repeated, are dependent upon their references to the original population data.

The fitness function for the trajectory GA rewards trajectories that exhibit the following characteristics:

- only existing generations are valid: all indexes are between the first and the last generation;
- the number of repetitions within the trajectory is related to the phrase population's overall complexity – more complex phrase populations can have a greater number of repetitions;
- repetitions are of shorter, rather than longer, phrase populations;
- repetitions are of more complex, rather than simpler, phrase populations;
- larger intervals (differences between generations) can occur for phrase populations that have higher overall complexity variations, so that when such a divergence occurs, it should be audible;
- backward intervals can occur for phrase populations that have higher overall density variations, so that such divergences are audible;
- phrase populations that are considered to be in the top 20% of those rated "interesting" are favoured – those that have the highest rated density, complexity, and similarity deviations.

The musical result of the trajectory GA produced a succession of generations that resembled those selected by hand: since the fitness function was based upon contextual information, the GA successfully operated at meso-compositional level.

3.4 Braiding: Combining Trajectories

In certain compositional instances, it is deemed musically desirable to vary and develop more than one idea during a section of music, often alternating between these ideas. In evolutionary terms, this corresponds to alternatively presenting two different trajectories (over two different sets of population histories), a process we call braiding. Just as the trajectory GA selects which generations to present from the original phrase population, determining when to switch between trajectory populations is a contextual decision entirely dependent upon the phrase populations. As such, another GA was created that utilizes the analysis described in Section 3.3, as well as the individual population trajectories calculated[1].

Individuals in this braided population consist of a binary switch, representing one of the two trajectories that are to be braided[2] – zeros represent successive selections from the trajectory of population A, while ones represent successive selections from the trajectory of population B (see Fig. 6).

Fig. 6 Two braided trajectories, with each dash – either a zero or one – representing the next succession in the trajectory population. The upper braided trajectory exhibits both short and moderate continuations for both populations, while the lower braided trajectory exhibits long continuations for population A, and short continuations for population B.

The initial population of 100 braided trajectories is generated using a $1/f$ (pink noise) function quantised to 0 or 1. A $1/f$ function was chosen since it resulted in longer continuations - a term we use to refer to the length of a continuous state of zero or one. Thus, the first five elements of the lower example in Fig. 6 are (0 0 0 1 1), which we consider to be two continuations. The first three elements indicate that the first three elements of population A's trajectory are to be used; the next two elements indicate the first two elements of population B's trajectory are to be used.

The fitness function rewards braided trajectories that exhibit the following parameters:

- the switch between populations (the change from 0 to 1 or vice versa) matches a change within the trajectory between generations;
- longer continuations occur for longer trajectory populations;

---

[1] In describing this GA, it becomes somewhat awkward in terminology, in that individuals in this braiding population reference individuals in the earlier described trajectory population, which in itself references generations of the original phrase population.

[2] At the moment, only two trajectories can be braided.

- the more complex and dense a trajectory's population, the shorter continuations it requires;
- greater variation in continuation lengths are required for those trajectory populations that have little variation in generation lengths;
- longer continuations occur at the beginning of the braided trajectory, unless that is where the longer generations are in the trajectory population;
- longer continuations contain shorter generations in the trajectory population;
- shorter continuations contain longer generations in the trajectory population;
- a balance exists between the two trajectory populations.

4. CONCLUSIONS AND FUTURE WORK

The system described successfully generates complete compositions that are representative of the first author's style, yet produce results that are original and/or musically interesting and surprising. Of note is the use of evolutionary algorithms to make high level musical decisions, which are dependent upon the context and content of the material.

At the time of writing this paper, the system has produced two complete compositions: a set of three virtuosic works for solo percussionist, and a work for percussionist, violin, and piano. When these works were presented in concert alongside a human-composed work by the first author, a formal audience survey confirmed that most listeners could not tell which pieces were computer-generated and which were not (this validation is the the topic of another paper).

Future work includes more research into pitch generation, which was not explored with the same rigor as rhythmic material. When pitch material was used in the second composition, it became evident that there was an abundance of motivic material, and that methods need to be developed to either tie together shorter motives, or else autonomously edit and reduce extraneous material.

Both compositions generated by the system are available on the first author's website, along with a video of the premiere performance.

5. ACKNOWLEDGMENTS

This research was funded by a grant from the Canada Council for the Arts, and the National Sciences and Engineering Research Council of Canada.

6. REFERENCES

1. Beyls, P.: Interactive Composing as the Expression of Autonomous Machine Motivations. In: Proceedings of the International Computer Music Conference. Montreal, 267-74 (2009)

2. Biles, J.: Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness. In: Proceedings of the 2001 Genetic and Evolutionary Computation Conference Workshop Program. San Francisco (2001)

3. Blackwell, T., and Young, M.: Swarm Granulator. Applications of Evolutionary Computing, In: Proceedings of EvoWorkshops 2004. Springer-Verlag, 399–408 (2004)

4. Brindle, R. S.: Beethoven’s Primitive Cell Structures. In: The Musical Times, 139:1865, 18–24 (1998)

5. Cross, J.: The Stravinsky Legacy. Cambridge University Press, Cambridge (1998)

6. Eigenfeldt, A.: The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine. In: Applications of Evolutionary Computing, Berlin, 498–507 (2009)

7. Ferneyhough, B.: Il Tempo della Figura. In: Perspectives of New Music, 31:1, 10–19 (1993)

8. Gann, K.: American Music in the Twentieth Century. Schirmer, New York (1997)

9. Haimo, E.: Schoenberg's Serial Odyssey: The Evolution of his Twelve-Tone Method 1914-1928. Clarendon Press, New York (1990)

10. McCormack, J.: Eden: An Evolutionary Sonic Eco-System. In: Advances In Artificial Life, LNCS. Springer, 133–142 (2001)

11. Miranda, E., Biles, J., eds.: Evolutionary Computer Music. London: Springer (2007)

12. Thywissen, K.: GeNotator: An environment for investigation the application of genetic algorithms in computer assisted composition. In: Proceedings of the 1996 ICMC, San Francisco, 274–277 (1996)

13. Todd, P., Werner, G.: Frankensteinian methods for evolutionary music composition. In: Musical networks: Parallel distributed perception and performance, Griffith, N., Todd, P., eds., Cambridge, MA: MIT Press/Bradford Books,. 313–339 (1999)

14. Warburton, D.: A Working Terminology for Minimal Music. In: Intégral, vol. 2 135-159 (1988)

15. Waschka, R.: Composing with Genetic Algorithms: GenDash. In: Evolutionary Computer Music, Springer, London, 117-136 (2007)

16. Weinberg, G., Godfrey, M., Rae, A., and Rhoads, J.: A Real-Time Genetic Algorithm in Human-Robot Musical Improvisation. In: Computer Music Modeling and Retrieval, Sense of Sounds, Springer, Berlin, 351-359 (2008)