Serendipity in Melodic Self-organising Fitness

Róisín Loughran and **Michael O'Neill** [1]

Abstract. Employing Evolutionary Strategies (ES) for subjective tasks such as melody writing causes an immediate problem in determining what to use as a fitness measure. By predefining a measure based on genre, musical rules or human opinion, as has been done in previous studies, we may be prematurely limiting the possibilities obtainable by the system, rendering serendipitous discovery impossible. In this paper, we discuss the development of a system that generates its own self-adaptive fitness measure in response to a corpus of evolved melodies. The system dynamically creates new fitness measures, or *Critics*, in response to new melodies in a cyclical manner with minimum human intervention. Thus it is a closed loop feedback system that develops its own fitness function through a response to its environment. We propose that the development of such a system could lead to more autonomous creativity and that the use of dynamically changing Critics and melodies could encourage the emergence of serendipitous discovery.

1 INTRODUCTION

Evolutionary strategies (ES) are driven by a fitness measure: a given measure or characteristic that determines an individual's likelihood to survive and reproduce. In a sense, this offers a parody to Darwin's theory of *natural selection* observed in nature, whereby individuals and hence species survive according to some survival traits emergent from the process of evolution in the real world. In computative ES, this fitness or survival measure is artificially pre-defined by the programmer to force the bred individuals to perform as they best see fit — to solve the given problem — which may be considered *unnatural selection*. Such specified goals and fitness remove any chance of serendipity in these systems; evolution moves towards the given goal without any regard for that which may be learned or discovered along the way. While this may have been shown to be an effective search method for traditional problems on which many ES methods were developed, such as symbolic regression or classification tasks, more recent methods incorporating alternative fitness measures and applications of ES methods to aesthetic domains have indicated that the field may encompass cybernetic methods, from which serendipity may be seen to emerge.

This paper presents the development of a melody generation system based on Grammatical Evolution (GE) named 'The Popular Critic' and discusses it from a cybernetic serendipitous point of view. While numeric fitness measures may be best for traditional ES experiments, such a measure is not simple to define for aesthetic applications; what number makes one melody better or worse than another? An overview of attributes used in the evaluation of melodies based on pitch and rhythm measurements is discussed in [7]. They conclude that previous approaches to formalise a fitness function for melodies have not comprehensively incorporated all measures. Some studies have addressed the problematic issue of determining musical subjective fitness by removing it from the evolutionary process entirely. GenDash was an early developed autonomous composition system that used random selection to drive the evolution [34]. Other studies only used highly fit individuals within the population from initialisation and then used the whole population in creating melodies [2, 9]. In the proposed paper, we consider a cyclical self-referential system that creates a fitness measure that responds to a corpus of melodies and then uses this fitness measure to create a new melody which replaces one of the existing melodies as the cycle repeats. Thus we create a melodic 'environment' that results in the response of creating a fitness which in turn alters the environment; the system results in a closed 'circular-causal' loop as postulated in early cybernetics studies. The discussion at the end of the paper reflects on the system as contributing to the study of cybernetic serendipity.

The following section describes some previous applications of ES to melody generation and the use of alternative non-traditional fitness measures used in evolutionary search. Section 3 describes the basics of GE and the workings of the proposed system. Section 4 presents some results obtained from experimental runs. Section 5 considers the system in terms of cybernetic serendipity and considers what implications it may have on future evolutionary strategies towards aesthetic applications and creative AI. Finally Section 6 offers some conclusions to the paper.

2 PREVIOUS WORK

ES refers to a family of algorithms that are all based on biological evolution including, but not limited to, Genetic Algorithms (GA), Genetic Programming (GP) and — as used in the proposed method — Grammatical Evolution (GE). Details of the workings of each of these systems can be found in [4]. As stated above, much previous work in ES has involved scientific experiments with standard, measurable numerical fitness. An excellent overview of ES systems applied to the aesthetic domains of art and music, specifically considering fitness measures is given in [14]. In this section we review some applications of ES methods to music generation before considering some alternative methods of measuring fitness in traditional domains.

2.1 ES in Melody Generation

Numerous EC methods have been applied to the problem of algorithmic composition. GAs have been applied in the systems GenJam to evolve real-time jazz solos [2], GenNotator to manipulate musical compositions using a hierarchical grammar [32] and more recently to create four-part harmony from music theory [12]. More recently, adapted GAs have been used with local search methods to investigate human virtuosity in composing with un-figured bass [24] and

[1] Natural Computing Research and Applications Group (NCRA), Michael Smurfit Graduate Business School, UCD, Dublin, email: roisin.loughran@ucd.ie

with non-dominated sorting in a multi-component generative music system that could generate chords, melodies and an accompaniment with two feasible-infeasible populations [29].

GP has been used to recursively describe binary trees as genetic representation for the evolution of musical scores. The recursive mechanism of this representation allowed the generation of expressive performances and gestures along with musical notation [6]. The first system to specifically use GE was proposed in [8]. In this paper GE generated melodies for a specific processor, although the melodies produced were not presented or discussed. GE has been implemented for composing short melodies in [27]. Interactive Grammatical Evolution (GE) has been used for musical composition with promising results [30, 27]. GE has also been used recently with autonomous fitness functions based on statistical measures of tonality and the Zipf's distribution of musical attributes [19, 18]. Zipf's distributions have been shown to correspond with aesthetics in musical compositions [22]. These studies found that the representation of the music created by the grammar and the combination of individuals from the final population could be as important as the fitness function. GE has also been proposed for generating a framework to produce live code in ChucK for use in real-time [21].

2.2 Serendipitous evolution

Rather than focussing on a pre-defined goal, the idea of searching specifically for *novelty* has proven to be an effective search strategy in evolutionary systems [17, 31]. This theory of 'novelty search' suggests that searching for novel solutions, never before seen by the system, rather than merely more fit solutions is a better method when considering a problem, as good or optimum solutions can be found when the search is not focussed on the goal. Such a theory is very apt when considering creative spaces and particularly when considering the concept of serendipity; novelty search considers the progress of the system and the space that has been considered and is not overly focussed on the current result and where it is in relation to a pre-defined goal. This is reminiscent of a creative act such as melody writing; a composer should not know their final composition from the outset, but consider the space they are working in and the evolution and development of their result at any time. We consider that for an automated evolutionary composition system to be creative there cannot be a pre-defined objective — the concept of progress and novelty must be considered, particularly to encourage the emergence of a serendipitous result.

Searching for novelty is dependent on previously observed outcomes within a given domain. A further consideration that may be taken into account in place of traditional goal searching is the search for *surprise*. Surprise differs from novelty in that it is dependent on an outcome that is different from that *expected* in a given domain. Surprise is based on expectation, which is based on inference from past experience, or on a temporal model of past outcomes. Hence, surprise can be viewed as a temporal novelty process. Surprise search has be proposed within an evolutionary system on creative tasks showing promising results [36].

An interesting study demonstrated that in Computationally Creative Evolutionary systems, there should be a move away from both random measure and pre-defined hard-coded fitness [5]. They propose that the most important aspect of a fitness measure is that it is defensible — not from a human subjective point of view but in a logical and reproducible manner. They create a logical fitness that is not based on human opinion but based on a series of comparisons resulting in sensible, defensible and reproducible choices by the program. This was investigated using the idea of a preference function by measuring specificity, transivity and reflexivity between individuals to determine the choices of a system in a a number of states. Such a system ignores the idea of human opinion in deference to the creation of an autonomous preference emergent from the system itself.

The environment created by the proposed method consists of a selection of melodies created by an earlier version of the GE system. The creation of these melodies is discussed in the following section. A population of 'Critics' are then evolved in response to this environment; there are complimentary evolutionary stages in the system but we would like to stress that this is not a co-evolutionary system. Co-evolution is an evolutionary system whereby two populations evolve in response to each other. A well-known musical co-evolutionary system based on bird-calls and responses has been proposed in [33]. The proposed system does not co-evolve melodies, however, but evolves Critics in response to a corpus of melodies which is then altered in response to the evolved Critic. The consensus of the population idea proposed here also shares conceptual similarities with the method in [23], which co-evolved agents with repertoires of melodies according to a measured 'sociability'. This sociability was measured in terms of similarity of the agent's repertoires; individual melodies survived or were altered depending on reinforcement feedback between co-evolving agents. This fitness differs from our proposed method as it is the correlation of a individual's opinion to that of the (single) population that is measured in this system rather than a direct similarity measure between melodies.

The system and terminology proposed in this study may also be reminiscent of the evaluation framework proposed in [26]. The proposed system differs in a number of important ways. This study does not attempt to conform to any particular style or genre of music but instead attempts to create an opinion among naive agents or 'Critics'. No indication as to whether the original melodies are good or bad is given. Furthermore, the proposed system is cyclical in nature, whereby the output is input back into the system for a dynamic evolution of further critics. Finally we do not include human evaluation or discrimination tests in our evaluation of the results, but instead focus on the diversity of the melodies produced. There is no aim towards human mimicry or trickery within this system.

2.3 Contribution of the paper

The purpose of this paper is to consider this evolutionary music generation system from a cybernetic serendipitous perspective. The system creates music, but while melodies are presented in Section 4, the focus of the paper is on the discussion and implications of the methods from a cybernetic perspective. The goal of such a system at its most simplest is merely to 'create music'; what may be discovered in the pursuit of such a generalised goal, while allowing the system to feedback to itself creating a sustainable closed-circular loop is the more interesting objective of this paper.

3 METHOD

There are three distinct phases to this compositional system:

1. The evolution of an initial musical corpus using GE;
2. The evolution of a Critic that conforms to the population's opinion as to which are the best melodies;
3. The evolution of novel music using this evolved Critic as a fitness measure which then replaces one of the original melodies in the corpus.

As the method is heavily based on GE [25], a brief introduction is given below.

3.1 Grammatical Evolution

GE is a grammar based algorithm based on Darwin's theory of evolution. As with other evolutionary algorithms, the benefit of GE as a search process results from its operation on a population of solutions rather than a single solution. From an initial population of random genotypes, GE performs a series of operations such as selection, mutation and crossover over a number of generations to search for the optimal solution to a given problem. A grammar is used to map each genotype to a phenotype that can represent the specified problem. The success or 'fitness' of each individual can be assessed as a measure of how well this phenotype solves the problem. Successful or highly fit individuals reproduce and survive to successive generations while weaker individuals are weaned out. Such grammar-based generative methods can be particularly suitable to generating music as it is an integer genome that is being manipulated rather than the music itself. This allows the method to generate an output with a level of complexity far greater than the original input. This added complexity generation is helpful in creating interesting and diverse pieces of music. In the system proposed, the grammar defines the search domain — the allowed notes and musical events in each composition. Successful melodies are then chosen by traversing this search space according to the defined fitness function.

We exploit the representational capabilities of GE resulting from the design of a grammar that defines the given search domain. GE maps the genotype to a phenotype — typically some form of program code. This phenotype can then be interpreted by the user in a predetermined manner. In this system, the programs created are written in a command language based on integer strings to represent sequences of MIDI notes. We design a grammar to create this command language which is in turn used to play music.

3.2 Creating the Musical Corpus

The Popular Critic is evolved according to its agreement with a population of its peers on their opinion of a selection of melodies. At initialisation, an initial corpus of 40 MIDI melodies was created using a previously developed system for composing short melodies with GE. This was initialised with previous melodies, instead of for instance know melodies, as this format can be used with the evolved Critics as described later. A full description of this method and the results obtained can be found in [19]. The following is an overview of the system. The grammar used is based on:

```
<piece>::=<event>|<piece><event>
 |<piece><event><event>
 |<piece><event><event><event>
<event>::=<style>,<oct>,<pitch>,<dur>
<style>::=<note>|<note>|<note>|<note>
 |<note>|<note>|<note>|<note>
 |<chord>|<chord>|<chord>
 |<chord>|<turn>|<arp>
<turn>::=<dir>,<len>,<dir>,<len>,<stepD>
<len>::=<step>|<step>,<step>
 |<step>,<step>,<step>
 |<step>,<step>,<step>,<step>
<dir>::=down|up
<step>::=1|1|1|1|1|2|2|2|2|2|2|2|2|3
```

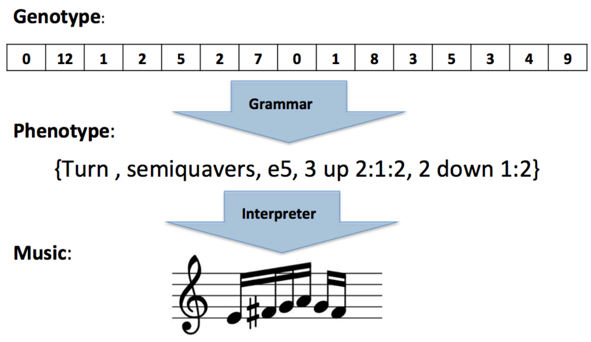

Figure 1. Application of Melody Grammar to integer Genotype through to representational Phenotype that can be interpreted into music.

```
<stepD>::=1|2|2|2|2|2|2|4|4|4|4|4|4
<oct>::=3|4|4|4|4|5|5|5|5|6|6
<pitch>::=0|1|2|3|4|5|6|7|8|9|10|11
<dur>::=1|1|1|2|2|2|4|4|4|8|8|16|16|32
```

This grammar creates a melody <piece> containing a number of notes with specified pitch and duration. Each <event> can either be a single note, a chord, a turn or an arpeggio. A single note is described by a given pitch, duration and octave value. A chord is given these values along with one, two or three notes played above the given note at specified intervals. A turn results in a series of notes proceeding in the direction up or down or a combination of both. Each step in a turn is limited to either one, two or three semitones. An arpeggio is similar to a turn except it allows larger intervals and longer durations. The application of this grammar results in a series of notes each with a given pitch and duration. The inclusion of turns and arpeggios allows a variation in the number of notes that are played, depending on the production rules chosen by the grammar.

This grammar is combined with the genotype to create the given phenotype — which can then be interpreted into MIDI note values. An example of this genotype to phenotype mapping for a short phrase is shown in Figure 1. This illustrates how a series of integer values can be transformed and interpreted in to a series of notes of specified pitch and duration through the applications of the above grammar. The selection of melodies into future generations is based on the defined fitness function. For this initial corpus the fitness is taken as a measure of the length of the melody combined with a statistical measure of prevalent tones within the piece. This is used to encourage the emergence of a pseudo-tonality (in that numerous pitches are repeated more often than others) but it does not enforce a key signature on any of the melodies. Initially the fitness is measured as:

$$fitness_{initial} = (Len - 200)^2 + 1 \tag{1}$$

where Len is the length of the current phenotype.

For an emergent tonality one pitch should be the most frequently played within the melody, with an unequal distribution of the remaining pitches. In the fitness the *primary* is defined as the pitch value with most instances and the *secondary* as that with the second highest number of instances. Thus for a good (low) fitness the number of primary pitches must be significantly higher than the number of secondary pitches. Furthermore, the number of instances of the seven most frequently played notes as Top7 and the number of instances of the top nine notes as Top9.

The fitness is multiplied by 1.3 if any of the following inequalities hold:

$$\frac{\#\ \text{instances of primary}}{\#\ \text{instances of secondary}} < 1.3 \quad (2)$$

$$\frac{\text{Top7}}{\text{Total number of played notes}} < 0.75 \quad (3)$$

$$\frac{\text{Top9}}{\text{Total number of played notes}} < 0.95 \quad (4)$$

This enforces the primary tone to have significantly more instances than the secondary and encourages most of the notes played to be within the top seven or top nine notes. These limits of 0.75 and 0.95 enforce more tonality than 12 tone serialism but will not create a melody with typical Western tonality. For these experiments, the top four melodies in the final population are concatenated together to encourage the emergence of themes within the final compositions. This grammar and fitness function create the corpus of 40 MIDI melody compositions which is then used to evolve the musical *Critics*.

3.3 Evolving the Critic

The purpose of this experiment is to dynamically design a new fitness function for adjudicating melodies that is not known to the programmer at the outset of the experiment. Our Critic is evolved to become the fitness measure to adjudicate the evolution of future melodies. This Critic (i.e. the fitness function) is itself evolved in the second phase of the experiment. GE is used to create this Critic as a specified linear combination of the content of the melodies.

The 'Popular Critic' is evolved by creating a population of individuals (or Critics), each of which gives a numerical 'opinion' of each of the melodies in the corpus. The melodies are represented as the number of times each degree of the scale and each note duration is played within the melody. Thus every melody is reduced to a list of 18 integer values. These instances are incorporated with a new grammar in GE shown below:

```
<expr> ::= <O><T1><O><T2><O><T3><O><T4>
   <O><T5><O><T6><O><T7><O><T8><O><T9>
   <O><T10><O><T11><O><T12><O><D1><O>
   <D2><O><D4><O><D8><O><D16><O><D32>
<O> ::= <op><scalar>
<op> ::= + | - | *
<scalar> ::= 1 | 2 | 3 | 4 | 5
```

This simple grammar takes each of the 12 tonal and 6 duration instances, multiplies each by a value 1-5 and then either adds, subtracts or multiplies it by the previous values. This outputs a scalar value resulting from a linear combination of the 18 given values. Each individual in the population results in a numerical value for each of the 40 given melodies. This is currently a *meritless adjudication* of the melody — there is nothing to say that 10 is better than 5 — it is merely a unitless numerical assignment.

In this system, however, we attribute 'preference' to this numerical output. The melodies are ranked 1-40 according to this numerical value, calculated by the given individual (the current Critic). These rankings are averaged across all individuals in the population and the overall ranking of the melodies across the population (of all Critics) is found. This overall ranking of all 40 melodies is taken as the popularity consensus of the population. The fitness of each individual Critic is then calculated according to how closely it correlates with this overall popularity, hence the fitness of the individual Critic is aligned with how much it conforms to the consensus of the population of Critics. The Kendall-Rank Correlation is used to calculate this fitness. Selection, Crossover and Mutation are then performed over successive generations to evolve one best 'Popular Critic' as with typical ES methods. The best evolved Popular Critic is saved to be used to evolve new music in the final phase of the system.

3.4 Critic-based Fitness

The Critic evolved in the previous section will output a numerical value for any melody that can be represented by the Melodic Grammar described in Section 3.2. As such, it can be combined with this grammar *as the fitness function* in a new, separate evolutionary run that will evolve the 'best' melody according to this given Critic. In the final phase of the system, we evolve a new melody and replace one of the original melodies from the corpus with this melody. This creates a change in the environment (the melody corpus) and the full system can be run again: using this new corpus (which differs from the original by just one melody), we initialise a new population of Critics to evolve a new Best Critic, which in turn can again be used in a new evolutionary run as a fitness measure to evolve a new replacement melody.

In this manner we have created a circular-causal loop, whereby Critics are evolved in response to their environment, which they in turn alter. Each cycle iteratively replaces one melody from the corpus. Once this cycle has repeated 40 times, all melodies in the original corpus have been replaced by those created by the system.

The following section discusses some results from various stages within the system. In all evolutionary runs we consider a minimising fitness. Each of the evolutionary phases were run with parameters, typical of GE runs, shown in Table 1, unless stated otherwise. An overview of the cyclical operation of the system is shown in Figure 2.

Table 1. EC parameters common to each evolutionary phase

Parameter	Value
Population Size	100
No. Generations	50
Selection	Tournament (size 2)
Crossover Rate	0.7
Mutation Rate	0.01
Initial Genome Length	100
Elite Size	1

4 RESULTS

While this section discusses numerical results obtained by the system, the interested reader may find a selection of melodies produced by the system available at `https://soundcloud.com/user-529879178/sets/serendipity-in-cybernetics`.

4.1 Fitness Results

The typical manner in which to judge an ES system is to consider the best and average fitnesses throughout the duration of the evolution. The best in generation and average in generation fitness in evolving the corpus melodies, averaged over 40 runs, is shown in Figure 3. This shows a fitness plot that is typical of a successful system, whereby both the average and best decrease initially and the

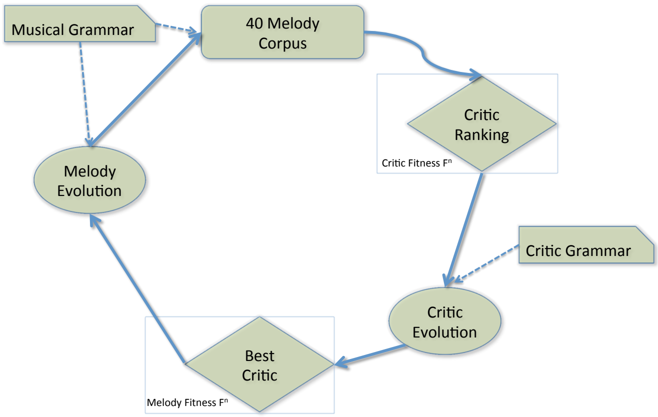

Figure 2. Overview of the cyclical system

best achieves a very good fitness by the end of evolution. The average fitness in the evolution remains less accurate as mutation and crossover are both kept until the last generation, to maintain diversity within the population. There is a strong drop off in the melody fitness around generation 10 (note the log_{10}) scale. This is because the fitness is initially taken in regards to the length of the phenotype, from Equation 1 which leads to large variations, before this is refined by smaller alterations due to Equations 2 to 4. These evolutionary runs may be considered successful as we observe the expected decrease in fitness measure, but that is merely because the individuals are being forced towards our pre-defined measures. Equations 2 to 4 are derived from *a priori* musical knowledge and theory, but they will not necessarily create the best music. This systems can evolve towards a given numerical goal but it cannot directly evolve towards any sense of musical beauty or creativity.

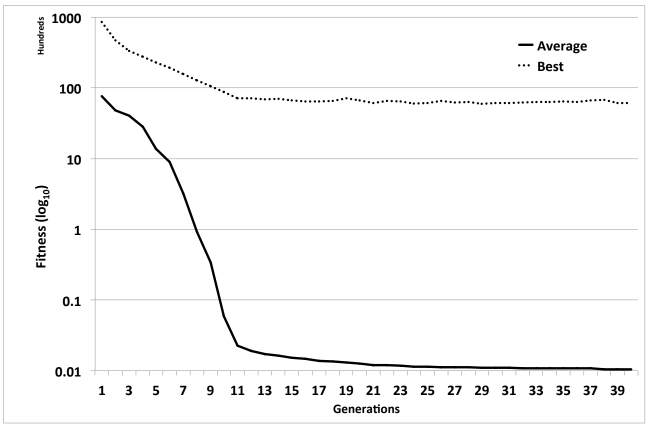

Figure 3. Fitness evolution of the melody corpus, averaged over 40 independent runs.

Similarly, we evolved 40 independent Critics according to the second stage of the system described in Section 3.3. The average of these results across 50 generations is shown in Figure 4. Again we can see the typical decrease in both best and averaged fitness throughout the evolution. The fitness values are notably smaller due to the method in which the fitness was measured. Again these measures do not necessarily tell us anything of the quality of the Critic — the Critics have not been evolved to be conventionally ‘good’ in any individual way but rather to conform to agree with each other; the relationship between Critics is more informative than the individual. As a measure of this we have considered the diversity within the generations of Critics throughout evolution. Even if two individuals result in the same fitness value, this does not mean their phenotypes are syntactically identical, this is dependent on the grammar. During the evolution of the 40 independent Critics we measured the diversity between the Critics at each generation. The population diversity was taken as the sum of the Levenshtein edit distance between the phenotypes of each pair of Critics. A plot of the average and standard deviation of this diversity is shown in Figure 5. This indicates a marked decrease in diversity within the first 10 generations (with a corresponding increase in standard deviation). Thus while the fitness is decreasing, on average, the diversity among the population is also decreasing. Again the level of mutation and crossover maintained throughout evolution means that the average fitness does not reach optimal as there is diversity left within the population. This is in keeping with what we had expected to see from the fitness plots, but in future experiments we will consider the temporal changes that occur within the Critic population in this respect.

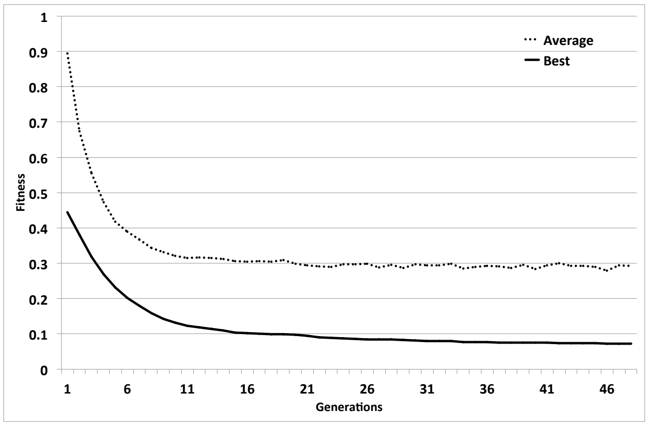

Figure 4. Fitness evolution of 40 Critics, averaged over 40 independent runs.

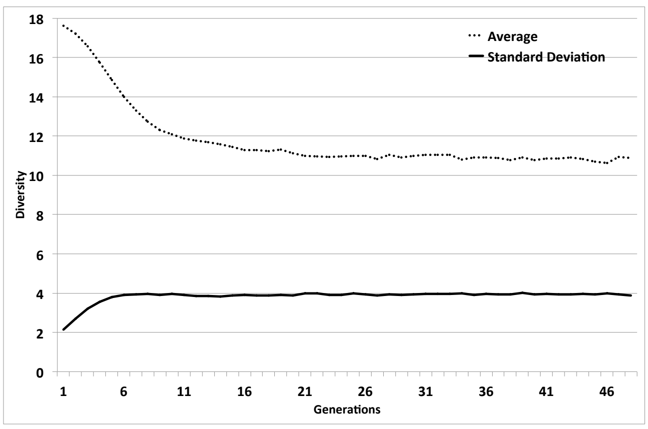

Figure 5. Diversity among 40 Critics, averaged over 40 independent runs.

4.2 Melodies

At the surface, the 'goal' of this system is to create melodies. It is the grammar used, however, that has most effect on the musicality of the system. The grammar developed creates the musical environment, which the Critics must traverse, and we have established that the individual Critics cannot distinguish musical merit but instead are evolved according to conformity to their peers. Melodies were created by the very first phase in generating the melodic corpus, so the development of the cyclical system purely for melody generation is superfluous for such a goal. To state that the goal is to improve these melodies is not realistic as the system stands; the Critics developed have no measure of human preference or musical theory embedded in them to create 'better' melodies. To judge any improvement in melodies from a human perspective is speculative — any improvement in this manner could only occur by chance. We instead consider the purpose of the system as a study in cybernetics within a creative domain, as further discussed below in Section 5. The content of the melodies is dependent on the grammar used to generate the individual phenotype, and the way in which this is interpreted by the program. We interpret each phenotype into a series of MIDI pitch and duration values that are then played through GarageBand using a MIDI piano sound. In listening to the melodies we can hear aspects of the grammar such as runs, arpeggios, chords and singles notes. The repetition of themes audible within the compositions indicate that the best individuals in the final population (the top four are concatenated to create each melody) are similar — but not identical. This indicates that the Critics are able to traverse a search space and converge on a stable idea. From a selection of melodies, it is clear that the system is capable of creating a wide variety of melodies. The selected melodies presented are a selection of the new final melodies created after a full cycle that display the different compositional elements of the system. For example Melody2 and Melody501 both display good examples of a mixture of runs and long notes, whereas Melody111 consists almost entirely of single held notes. This is because no specifications were made at any point during the cyclical system as to what constraints should be put on the melodies — the Critics are able to evolve to explore the full musical domain created from the genotype-phenotype mapping. While we do not focus on evaluating individual melodies at each cycle, it is worth considering the change in the melodies — or specifically within the corpus of melodies — as the system is run.

In each full cycle a new melody is generated which replaces one from the current corpus. As there is no meaningful adjudication as to which melody is best, the replaced melody is chosen iteratively from the corpus. Thus after 40 cycles, the corpus has been completely re-populated with melodies generated specifically by the system. If the system is allowed to continue to run, it will keep creating new melodies from the Critics that were created from the continually changing melodic corpus. We consider the diversity between the 40 melodies within the corpus after each full cycle. This was measured as the sum of the Levenstein distance between the representation (as 18 integer values — 12 for pitch, 6 for duration) of each pair of melodies within the corpus. A plot of this over 1000 consecutive cycles is shown in Figure 6. This plot shows an initial drop in the diversity among the melodies over the first 50 cycles. This implies that as the corpus is populated with melodies created by Critics emergent from the system, as opposed to those created initially, the content of the melodies begins to converge. Once the corpus has been repopulated, however, this trend does not continue over subsequent cycles. Instead we can observe a cyclical increase and decrease in diversity among the melodies as the system cycles. This is understandable when we consider that again, there is nothing within the generation of a Critic to enforce a homongenization of the melodies. It may be interesting in future developments of the system to determine if such a relationship could be enforced.

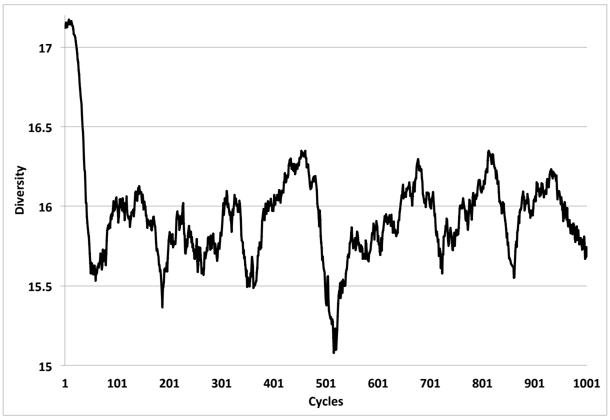

Figure 6. Diversity within the melody corpus, examined across 1000 successive cycles of the system.

5 CYBERNETIC SERENDIPITY

Cybernetics was first introduced as a theory based on the scientific study of control and communication in the animal and the machine [35]. It considers the manner in which a system behaves rather than mere results. Cybernetic systems have a closed circular or feedback loop, resulting in a 'circular-causal' relationship whereby system and environment are intertwined. Any action by the system generates some change in its environment and that change is reflected in the system in some manner (i.e. feedback) that triggers a system change, and this process repeats. During the development of the theory of Cybernetics in the last century, there appeared a split into two subfields: First Order Cybernetics — the study of observed systems and Second Order Cybernetics (or the Cybernetics of Cybernetics) — the study of observing systems. The split arguably grew from the increasing interest in engineering and computing systems which focussed on control (First Order Cybernetics) in contrast to those who wished to focus on autonomy and self-organisation (Second Order Cybernetics) [13]. Regardless of this split, the focus of Cybernetics has been on behaviour of the system; the question is not "what is the thing?" but "what does it do?" [1].

Cybernetic Serendipity was first coined through an exhibition curated by Jasia Reichardt, shown at the Institute of Contemporary Arts, London in 1968 [28]. This event showcased art and music created by algorithms and computers. The exhibition and subsequent publications were concerned with exploring the connections between art and technology. It was not considered merely an art exhibition nor a technology show, but a demonstration of contemporary ideas linking cybernetics and creative processes. The name was coined from the idea that in considering technological (particularly cybernetic) applications within artistic domains, serendipitous discoveries and developments would become apparent. The event showcased music, art, films and robotics to an audience of 60,000 attendees; it brought computer generated art and music to an audience that would never have before had access to such ideas. One of the most interesting concepts within the exhibition was that no artefact gave any indication as to whether, or to what level, it was created by man or machine. This aspect of human-involvement is still very important in computer generated creative artefacts today. Much debate remains on the merit of systems that exhibit purely generative behaviour as opposed to those that could be considered autonomously creative. Generative systems tend to have more human input, but as such are generally more sophisticated, exhibiting impressive results. Computationally creative systems, on the other hand, are those systems to which an attempt is made to attribute the creativity itself to the system, rather than the human engineer. Either of such systems would have been suitable for inclusion in this original Cybernetic Serendipity exhibition, although these days, the extent to which the creativity is displayed by the system is put to more scrutiny through the process of evaluation.

Evaluation of a cybernetic system should depend on whether or not it has achieved its goal. When considering aesthetic tasks, such as in 'serendipitous cybernetic' systems, this goal once again becomes difficult to define. Evaluation in computationally creative systems can be difficult to measure in a meaningful way; creativity itself is such a hard concept to measure, how can we reliably measure the display of it by a computer? This difficulty has led to a noted lack of evaluation in the development of computationally creative systems [3, 15]. This has been addressed with the development towards standardised measures of evaluation of creative system for e.g. the SPECS model [16]. In performing evaluations however, we must always ensure we are considering the true intention of the designed system. Some evaluations have only considered the output of the system — i.e. judging a melody generation system, such as the one above, purely on the perceived quality of the melodies produced. This assumes that the only purpose of such a system is to generate melodies that mimic how a human would compose melodies. Such assumptions could limit the possibilities attainable by these systems [20]. We do not yet know the capabilities of computative systems, if we limit their goal to merely aim to imitate what we already know, might we be limiting the capabilities of such system?

The focus of the original Cybernetic Serendipity studies was in the relationship between the arts and technology, and as such, some of the studies and artefacts may arguably be considered First Order Cybernetic systems. The 'Popular Critic' proposed in this paper is a conscious effort to consider a melody generation system that encompasses a circular causal feedback system. The system operates in a closed cycle; once it is set in motion it will continue without external input, continuously generating new melodies without any further human interference. While the environment (the melody corpus) is originally given, it alters this environment in response to its own interactions with it, within the confines of the grammars we have defined. As stated at the beginning of this paper, ES systems are dependent on the representation and the fitness measures used, but pre-defining a fitness measure for aesthetic problems is a difficult task. In this system we define two complementing grammars to define two levels of representation, but the fitness measure (for both the melody and the Critics) are changing fluidly in response to the workings of the system. Hence the notion of creating a fitness measure *a priori* that will ultimately confine the generated music to some pre-determined result can be avoided. Admittedly, this makes the goal of our system more difficult to define, but it simultaneously makes serendipitous discovery considerably more possible.

6 CONCLUSION

We have presented a cybernetic melody generation system focussed on the development of a self-adaptive fitness measure. The evolutionary system presented uses two complementary grammatical evolution runs combined into a cyclical system that generate both melodies and Critics. The system offers no measure as to what makes a musical melody 'good' but instead poses that a measure of agreeability, or popularity, among the population of Critics can be used to self-organise and autonomously generate melodies. The Critics are evolved in response to a corpus of melodies which in turn is changed by the evolved Critics and this cycle is repeated. In this manner, the Popular Critic operates as a circular-causal feedback system where Critics are created from and directly affect the environment in which they operate. We plan to explore using measures of aesthetic beauty such as fractal analysis [10, 11] in combination with the self adaptive methods prosed here in the future development of Critics. We have noted that the grammar used and its interpretation into MIDI messages are responsible for the musicality in the system. As such, we may implement a similar system in another aesthetic domain, such as visuals, to consider the possibilities of serendipitous discovery across multiple domains.

We believe the system as it stands is a good example of a melody generating Second Order Cybernetic system. However, we do recognise limitations in the practical application of the system. For those who are looking for good or pleasant sounding melodies, there is nothing in the running of this system that will ensure such a goal. The musicality of the system is completely emergent from the melody grammar used; melodies created after 1000 cycles of the system are likely to be as 'musical' as those from the original corpus. We acknowledge that we need a more clear and definitive method of evaluating the merit of this system in this manner. However, we also consider that this system is more interesting as a study in the development of autonomous fitness, particularly in an aesthetic domain such as music, where an ideal fitness could arguably never be defined from a philosophical standpoint. In this respect, this system offers a new method as to how we may consider using evolutionary computational methods in such domains. In future version of the system we are planning to continue with this method of emergent fitness, while considering more controlled ways of examining the workings and goals of the system. In taking the focus away from a pre-defined measure of pleasantness or goodness in music we hope to encourage more serendipitous emergence of new ideas.

ACKNOWLEDGEMENTS

This work is part of the App'Ed (Applications of Evolutionary Design) project funded by Science Foundation Ireland under grant 13/IA/1850.

REFERENCES

[1] W Ross Ashby, *An introduction to cybernetics*, Chapman & Hall Ltd, 1961.

[2] John A Biles, 'Straight-ahead jazz with GenJam: A quick demonstration', in *MUME 2013 Workshop*, (2013).

[3] Margaret A Boden, 'Creativity and artificial intelligence', *Artificial Intelligence*, **103**(1), 347–356, (1998).

[4] Anthony Brabazon, Michael O'Neill, and Seán McGarraghy, 'Grammatical evolution', in *Natural Computing Algorithms*, 357–373, Springer, (2015).

[5] Michael Cook and Simon Colton, 'Generating code for expressing simple preferences: Moving on from hardcoding and randomness', in *Proceedings of the Sixth International Conference on Computational Creativity June*, p. 8, (2015).

[6] Palle Dahlstedt, 'Autonomous evolution of complete piano pieces and performances', in *Proceedings of Music AL Workshop*. Citeseer, (2007).

[7] Alan RR de Freitas, Frederico G Guimaraes, and Rogério V Barbosa, 'Ideas in automatic evaluation methods for melodies in algorithmic composition', in *Sound and Music Computing Conference*, (2012).

[8] Alfonso Ortega de la Puente, Rafael Sánchez Alfonso, and Manuel Alfonseca Moreno, 'Automatic composition of music by means of grammatical evolution', in *ACM SIGAPL APL Quote Quad*, volume 32, pp. 148–155. ACM, (2002).

[9] Arne Eigenfeldt and Philippe Pasquier, 'Populations of populations: composing with multiple evolutionary algorithms', in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, 72–83, Springer, (2012).

[10] Alex Forsythe, Marcos Nadal, Noel Sheehy, Camilo J Cela-Conde, and Martin Sawey, 'Predicting beauty: fractal dimension and visual complexity in art', *British journal of psychology*, **102**(1), 49–70, (2011).

[11] Alex Forsythe, Tamsin Williams, and Ronan G Reilly, 'What paint can tell us: A fractal analysis of neurological changes in seven artists.', *Neuropsychology*, **31**(1), 1, (2017).

[12] Hüseyin Göksu, Paul Pigg, and Vikas Dixit, 'Music composition using genetic algorithms (GA) and multilayer perceptrons (MLP)', in *Advances in Natural Computation*, 1242–1250, Springer, (2005).

[13] Francis Heylighen and Cliff Joslyn, 'Cybernetics and second order cybernetics', *Encyclopedia of physical science & technology*, **4**, 155–170, (2001).

[14] Colin G Johnson, 'Fitness in evolutionary art and music: a taxonomy and future prospects', *International Journal of Arts and Technology*, **9**(1), 4–25, (2016).

[15] Anna Jordanous, 'Evaluating evaluation: Assessing progress in computational creativity research', (2011).

[16] Anna Jordanous, 'A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative', *Cognitive Computation*, **4**(3), 246–279, (2012).

[17] Joel Lehman and Kenneth O Stanley, 'Efficiently evolving programs through the search for novelty', in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 837–844. ACM, (2010).

[18] Róisín Loughran, James McDermott, and Michael O'Neill, 'Grammatical evolution with zipf's law based fitness for melodic composition', in *Sound and Music Computing Conference, Maynooth*, (2015).

[19] Róisín Loughran, James McDermott, and Michael O'Neill, 'Tonality driven piano compositions with grammatical evolution', in *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pp. 2168–2175. IEEE, (2015).

[20] Róisín Loughran and Michael O'Neill, 'Limitations from assumptions in generative music evaluation', *Journal of Creative Music Systems*, **2**(1), (September 2017).

[21] Róisín Loughran and Michael O'Neill, "My Little Chucky': Towards live-coding with grammatical evolution', in *Musical Metacreation (MuMe) (under review)*, (2017).

[22] Bill Manaris, Juan Romero, Penousal Machado, Dwight Krehbiel, Timothy Hirzel, Walter Pharr, and Robert B Davis, 'Zipf's law, music classification, and aesthetics', *Computer Music Journal*, **29**(1), 55–69, (2005).

[23] Eduardo Reck Miranda, 'On the evolution of music in a society of self-taught digital creatures', *Digital Creativity*, **14**(1), 29–42, (2003).

[24] Eugenio Munoz, Jose Cadenas, Yew Soon Ong, and Giovanni Acampora, 'Memetic music composition', *IEEE Transactions on Evolutionary Computation*, **20**(1), (February 2016).

[25] Michael O'Neill and Conor Ryan, *Grammatical evolution*, Springer, 2003.

[26] Marcus T Pearce and Geraint A Wiggins, 'Evaluating cognitive models of musical composition', in *Proceedings of the 4th international joint workshop on computational creativity*, pp. 73–80. Goldsmiths, University of London, (2007).

[27] John Reddin, James McDermott, and Michael O'Neill, 'Elevated Pitch: Automated grammatical evolution of short compositions', in *Applications of Evolutionary Computing*, 579–584, Springer, (2009).

[28] Jasia Reichardt and Jasia Reichardt, *Cybernetics, art and ideas*, Studio Vista London, 1971.

[29] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi, 'Metacompose: A compositional evolutionary music composer', in *International Conference on Evolutionary and Biologically Inspired Music and Art*, pp. 202–217. Springer, (2016).

[30] Jianhua Shao, James McDermott, Michael O'Neill, and Anthony Brabazon, 'Jive: A generative, interactive, virtual, evolutionary music system', in *Applications of Evolutionary Computation*, 341–350, Springer, (2010).

[31] Kenneth O Stanley and Joel Lehman, *Why Greatness Cannot Be Planned: The Myth of the Objective*, Springer, 2015.

[32] Kurt Thywissen, 'GeNotator: an environment for exploring the application of evolutionary techniques in computer-assisted composition', *Organised Sound*, **4**(02), 127–133, (1999).

[33] Peter M Todd and Gregory M Werner, 'Frankensteinian methods for evolutionary music', *Musical networks: parallel distributed perception and performace*, 313, (1999).

[34] Rodney Waschka II, 'Composing with genetic algorithms: GenDash', in *Evolutionary Computer Music*, 117–136, Springer, (2007).

[35] Norbert Wiener, 'Cybernetics', *Scientific American*, **179**(5), 14–19, (1948).

[36] Georgios N Yannakakis and Antonios Liapis, 'Searching for surprise', in *Proceedings of the International Conference on Computational Creativity*, (2016).