Genetic Programming and Evolvable Machines (2020) 21:55–85
https://doi.org/10.1007/s10710-020-09380-7

Evolutionary music: applying evolutionary computation to the art of creating music

Róisín Loughran[1] · Michael O'Neill[1]

Received: 20 October 2018 / Revised: 8 August 2019 / Published online: 6 February 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

We present a review of the application of genetic programming (GP) and other variations of evolutionary computation (EC) to the creative art of music composition. Throughout the development of EC methods, since the early 1990s, a small number of researchers have considered aesthetic problems such as the act of composing music alongside other more traditional problem domains. Over the years, interest in these aesthetic or artistic domains has grown significantly. We review the implementation of GP and EC for music composition in terms of the compositional task undertaken, the algorithm used, the representation of the individuals and the fitness measure employed. In these aesthetic studies we note that there are more variations or generalisations in the algorithmic implementation in comparison to traditional GP experiments; even if GP is not explicitly stated, many studies use representations that are distinctly GP-like. We determine that there is no single compositional challenge and no single best evolutionary method with which to approach the act of music composition. We consider autonomous composition as a computationally creative act and investigate the suitability of EC methods to the search for creativity. We conclude that the exploratory nature of evolutionary methods are highly appropriate for a wide variety of compositional tasks and propose that the development and study of GP and EC methods on creative tasks such as music composition should be encouraged.

Keywords Music composition · Evolutionary computation · Computational creativity · Review

✉ Róisín Loughran
roisis@gmail.com

Michael O'Neill
m.oneill@ucd.ie

[1] Natural Computing Research and Applications Group (NCRA), Michael Smurfit Graduate Business School, UCD, Dublin, Ireland, UK

1 Introduction

Throughout history, computers and music have exhibited a symbiotic relationship with one another. When presented with the Analytical Engine, a precursor to the modern computer, Ada Lovelace saw past the mere numerical possibilities and speculated on its ability to 'compose elaborate and scientific pieces of music' [70]. The first musical score to be composed by an electronic computer is generally agreed to have been the *Illiac Suite* in 1957 [35]. Just a few years later, in 1961 the IBM 704 at Bell Labs performed 'Daisy Bell' (or Bicycle Built for Two) as the first demonstration of computer speech synthesis [16, 53]. In 1968 the public was invited to 'Cybernetic Serendipity'—an exhibition curated by Jasia Reichardt which showcased art and music created by algorithms and computers [84]. As technology has developed, music has been used to showcase the abilities and possibilities attainable from the latest advances. The same is true in this age of Machine Learning (ML) and Artificial Intelligence (AI).

This paper presents a review of the application of GP and more generally EC to music, in particular to the creative act of algorithmic music composition. As the field of EC has developed in recent decades, the importance of creative applications such as music have become more apparent. The International Conference on Computational Intelligence in Music, Sounds, Art and Design (EvoMUSART) started as a workshop in 2003 as part of the EvoStar conference, and has now become a conference it its own right with independent proceedings since 2012 [41, 85]. Whereas back in the 1990s practitioners were writing conference papers on such topics, studies and discussions on the application of EC to creative challenges such as music and art have developed into journal articles, chapters and numerous full books focussing on these types of problems [74, 86]. In addition to academic papers, many of these systems have been implemented in live musical performances in jazz improvisations [4], the performance of arias [101] and piano pieces [19].

The upsurge of interest in aesthetic problems has come about as researchers realised that the automation of such processes pose much more of a challenge than the mere generation of 'pretty things'. Yes, EC systems can be used to produce music, but often the question of *how* they produce such music or artefacts can be more interesting than what they actually produce. Evolving aesthetic results has no one final solution, there is no goal to evolve towards. We shall see in this paper that many people have created EC systems that evolve music towards a set of music theory rules or a similarity to a corpus or an individual's taste and determined that these systems have 'successfully' created music. But as successful as these systems are, no one system can claim to be the best. Music composition is not a challenge that can be won; this is not a regression or classification task where the result can be measured. If this is the case—why do we apply ourselves to such tasks, and why has interest in this area increased over the years?

The question of why we apply EC to music generation can only be answered once we define exactly what it is we intend to achieve in undertaking this task. In this age of AI[1] there is much interest in mimicking the human brain, although many people maintain the belief that while computers can best a human at numerical challenges or strategic games, they would never be able to be creative. While the question as to whether or not a computer can genuinely be creative has not been definitively answered, it is a question that is now open for discussion. Unfortunately, as we detail in Sect. 5, this discussion starts with a lack of definition as to what creativity actually means. While the term *creative* may be colloquially understood, it is inherently difficult to define explicitly. In contradiction to many popular opinions, creativity is not a gift or talent that only a few people possess; creativity is merely an aspect of general intelligence [9]. If we acknowledge that creativity is an ability possessed by us all, it follows that creativity must be explored within the general field of AI. Because of this, the field of Computational Creativity (CC) has emerged and grown in recent years, supporting high-quality research into creative acts including story-telling, art and music. This interest had led to establishment of the International Conference of Computational Creativity (ICCC) [2]. In a parallel emergence of the EvoMUSART conference, ICCC has grown from a joint workshop started in 2006 into an independent international annual conference since 2010. Throughout the discussion of creativity and CC, it has often been noted that EC systems are particularly adept to being applied in this field [32, 46, 63]. We discuss this view, and the suitability of EC to the application of creative acts such as music composition later in this paper.

So when we ask 'Why apply GP to music composition?' we must consider what level of creativity we wish GP to emulate in this application. In the field of CC it has been proposed that the creativity emulated by autonomous systems is variable, along a scale from *mere generation* to *computationally creative* [29]. All systems along such a scale are useful in the understanding of CC, but to evaluate any system it is important to recognise where it lies on such a scale. Musical systems that focus on generation automate the process of composition and so it is the output of these systems that should be of high quality. Other systems are considered co-creative; such systems augment the creativity of the user in a computative manner. The evaluation of such systems is often based on user feedback. Often in systems that are deemed purely creative, the goal is to analyse the process of creativity; sometimes the primary function of such systems is not in creating something but in helping our general understanding of creativity. The concept of creativity itself is still not very well understood. GP can be applied to musical or creative systems in any of the above manners. In designing creative GP systems we must first decide if the goal is for the system to be creative or to help us understand creativity.

This paper offers an overview of the use of various EC methods applied to music composition from the 1990s up to the present day. Sects. 2 and 3 detail a number of such experiments, which are discussed more generally in terms of algorithm, problem, representation and fitness measure in Sect. 4. Section 5 discusses the definition and computer emulation of creativity and musicality and the use of EC on such applications as music composition. Finally, Sect. 6 draws some conclusions from the review.

[1] A term that is so over-used in the media it effectively refers to 'computer science'.

2 Evolutionary music

The application of EC methods to musical tasks began to emerge in the early 1990s, alongside other more traditional applications of EC. In applying EC to music there are a wide range of aspects to consider, however, the three fundamental criteria to consider upfront are problem domain, individual representation and fitness measure. The problem domain defines exactly what type of 'music' you wish to evolve; are you planning to create melodies, harmonisations, chord progressions? What style of music do you wish to create? In what format will it be played—through audio, printed scores, Musical Instrument Digital Interface (MIDI) messages? Once these high level decisions have been made you must decide how to represent your music as an individual. Does the genome represent pitch values, durations, are translational grammars required? The encoding of the domain, representation and operators are all intrinsically linked to define the space—and hence the musical limits or style—within which the algorithm searches. Finally there is the question of fitness: assuming two individuals represent pieces of music in your desired domain—what makes one 'better' than the other? One of the advantages of applying EC in this type of subjective domain is that it can optimise towards a relative measure of quality or fitness. EC continuously updates a population of solutions, rather than an individual, so the relationship between individuals can be incorporated into the search process. This is utilised in numerous studies that use the whole final population or multiple individuals from the population in forming a result. But, even if we are not evolving towards a single best objective, we do need some form of fitness measure to drive the search. This is one of the most challenging aspects of designing such experiments as will be discussed later on.

This section discusses numerous studies that have applied EC to various music composition tasks. Each study involves the application of either Genetic Algorithms (GA), GP or Grammatical Evolution (GE) or a variation of these to some aspect of algorithmic composition. We assume the reader is familiar with the general behaviour of these algorithms. Those new to the field may find an introduction to such methods in [10]. In each discussion we focus on approaches to problem domain, representation and fitness measures and less so on operators or other experimental specifics.

2.1 Early applications of EC to music

Horner and Goldberg [38] were one of the first to apply a GA to the process of musical composition by considering 'thematic bridging'. In this they considered the transformation of one melodic phrase to another through the use of a defined operation set. The individuals represented the series of transitions from one phrase to the desired phrase. Fitness was based on how well the final pattern matched the desired pattern. As such, the system was used to create the transitions (sequence of operations) rather than actually creating music. *NUREGEN* was a system developed to produce small four-part harmony melodies using an Artificial Neural Network

(ANN) as the fitness function [33]. The authors limited their domain by only considering harmonies based on three chords within the key of C major and only creating melodies of four bars in duration. The composition was then broken down into 'building blocks' consisting of rhythm, primary melody and harmony which were each composed in sequence. For each sequential block a GA was used to generate a population and an ANN was trained from examples given by the user to judge suitable candidates as the fitness function. The authors acknowledged that this combination of GA and ANN could produce pseudo compositions, but the limitations applied to the domain and the three levels of training input from the user limited the system.

Regardless of the representation used, music evaluation is most often a subjective task and, as such, a reliable and robust fitness measure can be difficult to define. Many early researchers circumvented this issue by employing the use of a human judge as the fitness function; such systems are known as Interactive EC (IEC). John Biles created the well known system *GenJam* which used a GA to evolve jazz solos [3]. GenJam used two independent populations: one for measures and one for phrases. Individuals in the measure population were mapped to a series of MIDI events; individuals in the phrase population were mapped to a series of measures. The user graded these phrases in real time as either 'good' or 'bad'. The system then used the entire population of phrases and measures to build a jazz solo. A human-based fitness function had been previously used successfully in evolving images [91] but the authors noted the practical difference in that multiple images can be viewed at once—and very quickly—whereas music is a temporal phenomenon and hence all individuals must be listened to sequentially in real time. This results in a very high cost in using a human as the fitness function, a problem termed the *fitness bottleneck*. While there is always some degree of fitness bottleneck in any interactive EC system, this issue is exacerbated in musical studies. Music is a temporal phenomenon and so it takes time to experience, let alone evaluate, the resultant outcome. This is similar to other applications requiring dynamic analysis such as automated programming whereby the result must be executed in some manner before a fitness measure can be made. For his music system, Biles addressed this bottleneck issue in later studies by training an ANN to act as the fitness function, thus removing the need for the human listener [5]. In this he found the results to be somewhat lacking however, and determined that humans listen to and experience music in complex and subtle ways that are not captured well by statistical models such as ANNs. GenJam has been modified many times over the last two decades and has been developed into a real-time, MIDI-based, interactive improvisation system that is regularly used in live performances in mainstream venues [4].

A system that generated musical rhythms using an interactive GA was proposed by Horowitz [39]. This system used a combination of methods along with user preferences in designing a fitness measure. Several objective functions were defined for which the user made specifications at the beginning of a run. The author noted that while the GA may be simple to implement, it is the fitness measure that is difficult to define and which ultimately determines the musicality of the output. A GP-like system was used for the automatic generation of style oriented music in an early system proposed by Laine and Kuuskankare [52]. In this Lisp system, melody lines were approximated by simple mathematical functions and the fitness was measured as a note by note comparison to a target input melody. The authors found that their biggest challenge was in finding functions that were complex enough to represent the full time-series of the music.

The problem of evaluating the work of 'constructed artists' was considered at length in the mid 1990s by Spector and Alpern [92]. In this paper the authors argued that at that time the AI community was concerned with rigour and standards in such systems and many were calling for the use of standard examples and more criteria for assessment.[2] They rejected the formerly used assessment methods based on human judgment or formalised criteria and instead proposed that the 'right' criteria were not strictly necessary for such judgments; a range of criteria was sufficient for adjudication if it was known that the system was capable of conforming to the given range of criteria. They proposed a system framework based on GP that would produce an artist—one that could produce new artworks that were successful to the given criteria and a culture defined by a given library of past works. They illustrated the framework by generating a BeBop musician that created jazz melodies. This system generated four-measure melodies as output when given four-measure melodies as input. The melodies were represented as a list of 64 numbers and the function set consisted of 13 distinct functions. The fitness was measured from a number of criteria that were compared against all of the melodies from the case-base when the program (the individual) was run. From their results they noted that the produced melodies that were most pleasing to the system were not the most pleasing to the authors and that although their framework did not rely on any particular critical criteria, it did require some encodable criteria. This system was extended by using connectionist techniques trained on a corpus of music to make judgements on the quality of the melodies produced [93]. This GP system used a more generic function and terminal set. The fitness was measured by a three-layer network trained to recognise reasonable continuations to reasonable fragments of jazz melody. Spector and Alpern found that combining the two types of fitness measures into a 'hybrid critic' offered the best results.

A GA compositional system was proposed by Jacob that considered the problem of the search domain, not by reducing the size of the domain, but by employing larger building blocks for the GA to work with [42]. The system was implemented by reducing the compositional process to a number of simple rules based around a set of primary motives. The composition, evaluation and arrangement of the music was then performed by agents with three specific modules—the composer, ear and arranger. A human observer judged the modules' ability to perform its given task and recombined successful agents for future generations. Instead of working with individual notes, this system concentrated on generating short phrases and combining these phrases into larger pieces of music. Jacob described this system in more detail amid a more general philosophical look at algorithmic composition as a model for creativity in [43]. In this paper he stated that there were two types of creativity:

[2] Over 20 years later this is still an issue, as we discuss below.

inspirational or genius and hard work. We contest this view of creativity below in Sect. 5.

An Adaptive Resonant Theory (ART) neural network was used as the fitness measure in a GA applied to musical rhythm composition by Burton and Vladimirova [13]. The ART network used unsupervised learning to cluster similar patterns in accordance to a given vigilance. Initial parameters such as rhythm type, tempos and time signature were given while the representation consisted of a binary array of data. Fitness was based on the similarity of the produced rhythm to an appropriate rhythm pattern cluster. The authors concluded that the ART network possessed an advantage over other neural networks for this purpose of acting as a fitness for a GA as they had the ability to add new classifications when the existing clusters could not represent the given individual.

During the 1990s many practitioners applied EC to aspects of music generation, but no one ideal method has emerged as the best choice. Burton and Vladimirova offerred a comprehensive overview of EC applications to music during this time [14]. They determined that GP techniques were generally more successful than GAs at compositional tasks, due to their ability to work in a less constrained space.

3 Twenty years of GP music

In keeping with the objective of this volume, we present a more detailed look at how GP, and all variants of EC, have been applied to music creation from 1998 to 2018. An overview of these systems is given in Table 1.

GPMusic was an interactive system developed by Johanson and Poli which allowed users to evolve short musical sequences using interactive GP [44]. To tackle the fitness bottleneck, users ratings were used to train a neural network to be employed as an 'auto rater' (automatic fitness measure) for longer runs. The musical sequences were represented as a sequence of musical notes in the extended module (XM) format. The terminal set consisted of notes, allowed chords and rests and the function set was made up of operations that could be performed on a given note sequence. In this manner, an evaluated individual program generated a string of notes. A user was then asked to rate the resulting musical pieces. Interactive studies discussed runs with increasingly complex functionality added to the GP run. Finally a run of 20 individuals over 10 generations was used to gather training data for the ANN. Three trials using this trained ANN indicated that the system could evolve interesting, pleasant music but it would not achieve the consistency shown by the human user.

Todd and Werner [98] proposed a 'Frankensteinien' imagery to the use of EC for algorithmic composition:

> But when, with Frankensteinian hubris, we dare to create an artificial system and imbue it with the spark of musical invention in our stead, how are we to assemble its constituent parts to ensure that its behavior will be on the whole pleasing and majestic, rather than filling us with aesthetic horror that no mortal could support?

Table 1 Overview Of EC applications to music composition

System	Problem	Alg.	Representation	Fitness
Horner [38]	Thematic Bridging	GA	Transitions	Target comparison
NUREGEN [33]	4-part harmony	GA	4 bar melodies	ANN
GenJam [3]	Jazz improv.	GA	MIDI	Interactive/random
GenJam [5]	Jazz improv.	GA	MIDI	ANN
Horowitz [39]	Rhythms	GA	Rhythm sequence	Multiple criteria
Laine [52]	Melody line	GP	Mathematical	Target comparison
Spector [92]	Jazz melodies	GP	4 bar melodies	Multiple criteria
Spector [93]	Jazz melodies	GP	4 bar melodies	ANN
Jacobs [42]	Building blocks	GA	Building blocks	Interactive
Burton [13]	Rhythm	GA	Binary array	ART NN
GPMusic [44]	Short melodies	GP	XM	Interactive/ANN
Todd [98]	Melodies	GA	Bird pairs	Co-evolution
GeNotator [96]	Melody	GA	MIDI	Interactive
Pearce [79]	Drum and bass	GA	Drum loops	MLP
Vox Populi [75]	4-part melodies	GA	4 words of 7 bits	Interactive
De la Puente [24]	Melodies	GE	Note values	Multiple criteria
Music Blox [31]	Music composition	GA	3D visual blocks	Similarity to target
Manaris [65]	Music composition	GP	MIDI	Zipfs Law
NEvMuse [64]	Music composition	GP	MIDI	ANN
Dahlstedt [20]	Scores/sound synth	GP	Recursive trees	Statistical measures
Khalifa [49]	Music composition	GA	16 gene motif	Grammars/rules
Phon-Amnuaisuk [80]	Melodies	GP	Musical trees	SOM
NEATDrummer [37]	Drum tracks	EC	MIDI drums	Interactive
GenDash [102]	Music composition	EC	Measure of music	Random
Elevated Pitch [83]	Short melodies	GE	MIDI	Interactive/rules
De Prisco [25]	Unfigured bass	GA	i/p harmonisation	Multi-objective
Donnelly [28]	4-part harmony	GA	4-part tuples	Scores
McDermott [69]	Music composition	GP	Executable graphs	Interactive/features
Melomics [27]	Auto. composition	GA	Unspecified	Multiple criteria
Jive [90]	Music composition	GE	Wii remote	Interactive
De Freitas [22]	Music composition	GA	Music measure	Random
DarwinTunes [61]	Musical loops	EC	Tree-based structure	Interactive
Eigenfeldt [30]	Music composition	EC	Musical motives	Multiple criteria
Pirnia [81]	Melodies	GP	Trees	Target based
Sulyok [95]	Compositional process	GP	Program	Statistical (Bach)
Kunimatsu [51]	Blues composition	GP	Chords/melodies	Musical/entropy
PIEC [97]	Phrase imitation	GA	Phrase notes	4 rule-based
Hofmann [36]	Music composition	GP	Constraints	Muli-objective
Loughran [56]	Melodies	GE	MIDI	Statistical
MetaCompose [89]	Music composition	GA	Note values	Multi-objective
Muńoz [76]	Unfigured bass	MA	Bass line	Harmonic rules
Loughran [55]	Melodic shapes	GE	MIDI	Target shape

Table 1 (continued)

System	Problem	Alg.	Representation	Fitness
Loughran [59]	Melodies	GE	MIDI	Critics/popularity
Plecto [40]	Audio synthesis	GA	Audio	Interactive
Hickinbotham [34]	LIve-coding	GA	Tidal code	Interactive
Loughran [57]	Live-coding	GE	Chuck code	Random
Olseng [78]	Music composition	GA	Melody and harmony	Multi-objective

They discussed multiple applications of EC to musically creative tasks but focussed on their coevolutionary system of evolving both the music and the critic (or fitness measure) simultaneously. From the inspiration of bird-song, they evolved male agents who played musical melodies along with female critics who judged the performed melodies. The females chose a mate according to which melodies they liked best. In their preliminary experiments both males and females were coded as neural networks that were evolved over time; the males produced rhythms and the females chose a mate according to how much she liked or disliked the song. A later experiment replaced the song-generating network of the male with a set of genes that directly encoded the melody. Each male song consisted of 32 notes whereas each female's genes encoded a transition matrix that rated transitions from one note to another in the male's songs. Each female listened to all male's compositions and chose one as her mate, ensuring that all females had one mate but males may have had multiple, or no, mates. Each mating pair had one child per generation before the population was again reduced to its target size. Three types of scoring by the females was considered. The authors concluded that while the system created the musical diversity and novelty they desired, the final produced melodies were lacking in musical structure and not pleasant to listen to.

GeNotator was a composition tool proposed by Thywissen that used a modified GA to manipulate a musical composition using a hierarchical grammar [96]. The system combined a user-defined music grammar with an automatically generated genetic description to map individual genes in the chromosome to choices from the grammar. This was input to an interactive GA. The phenotypes were played as music by generating a MIDI stream using the grammar. Originally the grammar had to be specified in textual form, but the author developed a graphical way of specifying a grammar making GeNotator more accessible to those not versed in computer programming. The author acknowledged that the system can create interesting music, but that this is often dependant on a good starting position and a level of form space bounding. While GeNotator may have been based on a GA, the developer made a number of amendments from the standard form in developing this system.

Vox Populi was an interactive compositional system developed by Moroni et al that evolved four-part melodies towards harmonical compatibility or a tonal centre [75]. Each individual had a genotype represented as a chromosome of four words of seven bits, each word representing a voice. The phenotype was the corresponding chord. The user had graphical control over the fitness measure which were based on the melody (tonal centre used to evaluate fitness), biology (time given to evolutionary operators), rhythm (time between evaluations), octave and orchestra. The authors concluded that Vox Populi could generate complex sound structures with perceptual and efficient control in real time.

When proposing a framework for evaluating machine compositions Pearce and Wiggins [79] illustrated the framework on an evolutionary Drum and Bass system. The authors proposed four essential elements in an evaluation framework: specifying an aim; inducing a 'critic' from a corpus of data; composing music; and evaluation. The example system employed a GA to create drum loops using a trained ANN as the fitness measure. While we may not necessarily concur with the details of each step—such as requiring a corpus to develop a critic (or fitness function), this was an important step in formalising the application of EC to music composition.

De la Puente et al were the first to apply GE to the problem of algorithmic composition [24]. In this paper GE generated melodies for the AP440 auxiliary processor. The melodies were encoded from pitch and duration values into a vector from which the fitness was measured. The melodies produced were not presented or discussed.

Music Blox was a real-time compositional system proposed by Gartland-Jones that utilised a domain specific, knowledge rich GA [31]. The system was created as a way of defining a limited search space for a non-composer user and allowing that space to be explored by the user. The system used blocks, depicted in a 3D visual graphical model, each of which had the capacity to play music. The blocks were then combined together in a physical structure, which in turn created a piece of music. The organisational structure of the blocks controlled the music played. Each block consisted of a bottom-up and top-down compositional system. The starting music fragments or 'home' music for each block was provided by the user. This was then directed towards a specific target melody using a GA. The fitness measure was dependent on a similarity measure between the home and target fragments. The author proposed that this system thus addressed the fitness bottleneck by developing an automatic fitness measure and considered the issue of over-limiting the search space. Gartland-Jones and Copley described the reasoning behind the system along with a general consideration of the usefulness of GAs to the challenge of musical composition in [32]. They described GAs as a directed search process and discussed how creative search processes relate to music composition.

Zipfs Law is a law often observed in nature relating to the statistical frequency of occurrence of events. It states that the frequency of an event is inversely proportional to its rank in frequency of occurrence [105]. For example, in a Zipfs distribution of pitches in a piece, if C is the most popular note followed by G and A, there will be twice as many Cs in the piece as Gs, and three times as many Cs as As etc. Such laws have been used in the investigation of pleasantness in music [66]. Musical events such as pitch, duration, melodic intervals and melodic/harmonic bigrams have all been shown to follow Zipfs distributions to varying degrees in numerous genres of music. The ideal of this Zipfs distribution was used as the fitness measure to drive an evolutionary composition system by Manaris et al [65]. This system was based on the NEvAR system that evolved visual art using symbolic representation in GP [62]. This idea was further developed by using the Zipfs distributions to train a neural network which was then used as the fitness measure in a GP compositional system entitled NEvMuse [64]. More recently, Loughran et al used the Zipfs distribution among a number of melodic and rhythmic aspects of music in the fitness measure in a system that used GE for the composition of MIDI melodies [54].

A number of evolutionary methods for generating novel musical scores and synthesised sounds have been proposed by Dahlstedt [20]. Dahlstedt's thesis described the underlying evolutionary methodologies and a number of interactive installations, tape and concert pieces that were composed and performed using these methodologies. He later described the composition of complete piano pieces and subsequent performance of these pieces [19]. This system implemented recursively described binary trees as genetic representation for the evolution of musical scores. In an individual tree each leaf represented a note or a series of notes while the branching nodes contained operators that merged or concatenated notes into larger segments. The trees were recursive in that a leaf node could contain a pointer to a branching node higher up the tree. Crossover and three types of mutation were used as operators. The system could be initialised randomly, with genomes previously stored or from human input. He noted that this representation contained no information about keys or harmony and that the recursive form could quickly create interesting musical fragments, making fitness evaluation difficult. A number of statistical measures were taken to eliminate undesirable characteristics. A weighted sum of these measures were used as the fitness measure.

Khlaifa et al developed an autonomous music compositional system using GAs with the integration of formal grammars [49]. This system started with the development of 16 musical motifs, whereby each individual motif contained 16 genes which allowed up to 16 notes in each motif. A grammar based fitness function was used to evaluate these motifs. The production rules of the devised context free grammar were based on known musically pleasing relationships between notes and chord progressions according to Western tonality. In the second stage of the system, these motifs were combined to form longer phrases. The fitness measure for these combined phrases was based on a combination of a measure from the acceptable intervals between notes and a relationship from the ratio of notes within the composition. The authors determined that interesting music could be composed but that multi-objective optimisation would be beneficial to the system.

Phon-Amnuaisuk et al proposed a method that used a Self-Organising Map (SOM) as a fitness function with GP for evolving simple melodies [80]. Each individual was made up of branching functions at the top of the tree with musical functions which occurred below them. Each musical function acted on an automatic defined function (ADF) which defined the note to be played. The SOM was trained on well known simple melodies. The authors proposed that the SOM provided a flexible means to capture domain knowledge while the use of GP searched around the given examples. ANNs were also used in combination with an evolutionary approach in the NEAT Drummer system [37]. This system was based on the NeuroEvolution of Augmenting Topologies (NEAT) system that evolves the topology of an ANN [94]. NEAT Drummer applied these principles to the generation of drum loops with the user interactively selecting towards MIDI rhythms of their choosing.

In contrast to more traditional applications of EC, some musical applications avoid the fitness problem altogether by implementing the algorithm with random fitness. It is arguable as to whether or not such a system actually *evovles* but if the focus is on the aesthetic output, and the representation used is strong and musical, a random fitness measure can be used to good effect. GenDash was such a system that used random fitness but also used and played all individuals that were created as music [102]. The author detailed the use of GenDash for the particular challenge of composing a single line of vocal music and the performance of the composed aria 'Sappho's Breath' in 2001 [101].

Reddin et al developed a system called Elevated Pitch which employed GE for composing short melodies using four different experimental set-ups of varying fitness functions and grammars [83]. In this system the authors used a combination of automatic fitness generation and interactive human judgment on a number of methods. They determined that users preferred melodies created with a structured grammar.

A multi-objective GA was proposed by De Prisco et al to the problem of unfigured bass harmonization [25]. In such problems the composer is given a bass line and must compose the three other parts to create a four-part harmony. In this system, the individuals were harmonizations of the input bass line and the two objective functions were based on harmonic considerations and melodic considerations. The authors tested the system using Bach chorales, measuring the number of melodic and harmonic errors. Four part harmony was again considered using a GA by Donnelly and Sheppard starting from a single musical chord [28]. Each individual contained four parts, one for each voice, consisting a list of tuples representing pitch and duration. This system evolved melody, harmony and rhythm all at once; the fitness of an individual was a weighted combined measure of a number of scores from rules for each of these three aspects of the composition. In their experiments the authors examined the probabilities of their operators, determining that by the final generation, crossover accounted for most of the operations. They determined that their system could create four-part compositions that displayed a number of desirable musical qualities.

Melomics (Melody-genomics) was an approach to algorithmic composition based on evolutionary techniques that was focussed on the complete automation of the composition of professional music [27, 82]. The fitness functions used with Melomics were developed from a collaboration with professional musicians and assessed the generated compositions according to various criteria of formal and basic aesthetic nature. Melomics was implemented in two computer systems: Iamus produced scores of complete contemporary works and Melomics109 was a computer cluster dedicated to composing and synthesizing popular music. Although details of the system are unclear, it is stated that it was based on an evolutionary-developmental process where evolutionary changes interpreted as small mutations in the genome could create complex changes in the phenotype.

GE was used with an interactive fitness function for musical composition using the Wii remote for a generative, virtual system entitled Jive [90]. This system interactively modified a combination of piece-wise linear sequences to create melodic pieces of musical interest. This Jive system and the NEAT Drummer were used to influence a system from McDermott and O'Reilly [69] that generated music based on executable graphs. This system implemented a representation where a piece of music consisted of a directed, acyclic multigraph along with several continuous control variables. They ran the system with three separate fitness functions: one based on 24 musical features, measurements from the variety in the graph's behaviour and an interactive function.

De Freitas and Guimarães [22] presented a system that considered the evolution of a composition from a known piece to repeated patterns with a random fitness measure. In this system each individual represented a measure of music and the full population of measures formed the melody. While there was a minimum implicit fitness evaluation in the generation of melodies, the evolution was not driven by an explicit fitness function. Instead this system used random fitness and observed how long it took for genetic drift to result in takeover (whereby all or most individuals in the population are identical). In this case, when takeover occurred, the population translated into a melody with a constant repeating measure. The system was initialised with a known melody and evolved until takeover occurred. The authors considered that the most interesting melody would be neither at the end of evolution (where the melody is too repetitive) nor at the beginning (where the melody is too similar to the original input) but somewhere in between. This point was selected as a measure of the originality-diversity trade-off calculated from the devised *Takeover Matrix*. Each generation was assigned a *compromise* value according to the Takeover Matrix. The median of the compromise values could be used to select the generation which should be used as the final resultant melody. They concluded that the system could be used to either generate melodies or extend and develop earlier known melodies.

DarwinTunes was a system developed for studying musical evolution in which fitness was determined by the aesthetic tastes of a group of musical consumers [61]. Each individual consisted of a tree-based structure, representing a program that generated a musical loop. Each population contained 100 loops each of which were 8 s long. When 20 loops had been rated, by the user on a five point scale, truncation selection allowed the top 10 loops to reproduce. This system involved large scale Interactive EC whereby 6,031 consumers made 85,533 ratings over 2,513 generations. Using both interactive web-based evaluations and two music information retrieval (MIR) algorithms, the authors determined that the evolved melodies improved over the first 500-600 generations and then fluctuated around a long-term mean. Although they stated that this could 'shed light on the evolution of real musical cultures' they did acknowledge that music in society is shaped by other forces than merely liking a loop. Cultural, social, societal, historical and peer influences are among many factors that are likely to effect the evolution of any real music, but no such measures were considered in this experiment.

Eigenfeldt and Pasquier [30] considered the time-based nature of music in their system in which individuals within the population represented musical motives and the audible evolution of populations over time were of musical interest. Musical designs and considerations within the system were influenced by the author's compositional style. Throughout evolution, the population remained ordered and the temporal sequence of all individuals in the population constituted a musical phrase. A trajectory was determined according to the variation over generations, which was then used to select the order and repetition of generations over time. Trajectories were selected according to a fitness function that rewarded certain characteristics. These trajectories could then be combined or 'braided' to form full compositions. The authors concluded that the system generated complete compositions that were musically interesting while being representative of the composers's style.

Pirnia and McCormack [81] examined representations of musical structure and their suitability for use with EC methods. They designed a series of experiments with a target based approach: for each representation, the musical space was searched towards a variety of pre-defined target melodies. The fitness measure was based on edit distance from candidate to target melody. They examined three representations: standard GP with eight functions and notes as terminals, extended GP which could use both notes and durations and ADFs which were able to use re-usable components. They found that the first representation suffered from the fact that the pitch and duration were tied together as one 'note' structure. The GP with ADFs representation were found to be most effective in finding melodies that had high instances of repetition but the addition of ADFs did not help significantly in melodies that had fewer repeated patterns.

3.1 Recent trends

GP was employed by Sulyok et al to examine evolving not a musical piece but a compositional *process* [95]. The action of composing the piece was considered as a process running on a Turing-complete virtual machine. Each individual genotype was run on the virtual machine, the output of which was parsed by a model builder to give a musical model. Each musical model was then evaluated in relation to a corpus of Bach melodies according to a number of statistical properties. This mechanism operated in a similar manner to grammar based evolutionary approaches where the grammar is replaced by the virtual machine. Kunimatsu et al also used GP in constructing two cooperative models for chord progression and melody [51]. In considering Blues music, they implemented a GP tree structure where the depth of the tree corresponds to the length of the note. The system composes by generating a melody followed by chords and finally a bassline. The fitness measures are developed from a comparison of the notes within the chords and examining the entropy function from partial progressions

Ting et al proposed the phrase imitation-based evolutionary composition (PIEC) system which used a GA to create a new melody from the characteristics of a given melodic phrase [97]. The system performed intraphrase and interphrase rearrangement to control phrase motion and fixed inappropriate notes to reduce dissonant intervals. Four rule based fitness functions were designed: difference of note variance, difference of interval variance, rules of arrangement and a hybrid method. The authors proposed that by imitating phrases, PIEC could create compositions similar to the sample melodies in characteristics without merely copying notes within phrases.

Hofmann proposed a GP approach to musical composition, written in a domain-specific language, by representing musical pieces as a set of constraints changing over time [36]. The constraints included tonal, rhythmical, instrumental and musical elements that control the composition of the music. Individual tree-based structures consisted of musical constraints, constraint modifiers, constraint generators and control structures. The model was modular, allowing subtrees to be referenced so that musical ideas could be re-used and varied within the composition. The system used a multi-objective fitness from a large range of statistical musical measures along with a number of structural features of the model. The authors noted that while their successful compositions were able to meet the statistical requirements of their fitness functions, the results were not always aesthetically pleasing.

Loughran et al used GE for the composition of short MIDI piano melodies [56]. For this system they developed a grammar that created individuals consisting of musical elements such as notes, turns chords and arpeggios. The fitness of the individual was measured, not in relation to Western tonality but from a statistical distribution of the pitches in each melody. Final compositions were formed by concatenating the top four individuals in the population. As these evolved towards a common goal, they were found to be similar, resulting in the emergence of motifs in the composition. The similarity between these individuals or motifs was used to evolve a later version of this system that aimed to match the distance between the individuals to given patterns [55]. A further development of this framework attempted to remove the explicit fitness measurement and instead investigated an implicit measure of fitness evolved as separate population of critics [59]. In this system a corpus a melodies was created, as per the original GE system. A population of critics were initialised that could give a numerical value for each melody. While this had no individual value as to the merit of the melodies, these critics were then evolved according to a mutual agreement as to which were the better or worse melodies in the corpus; the critic that had the most popular opinion received the best fitness. The best evolved critic could then be used as a fitness measure to evolve a new melody for the corpus and the cycle repeated. The authors noted that this would not necessarily create 'better' melodies than their original system, but it did create a complex adaptive system that would generate musical output without relying on an explicit fitness measure from the user. An extension of the system that examined clustering of melodies rather than pure numerical rankings has been considered [58].

MetaCompose was an evolutionary music composer that split the act of composition into three processes: chord generation, melody generation and accompaniment [89]. The chords were generated using a random walk through a directed graph of common chord sequences and the accompaniment was generated with a stochastic process. The melody was generated using a multi-objective optimization evolutionary technique based on NSGA-II [26] with constraints of two Feasible/Infeasible populations as used in FI-2POP [50]. The fitness was measured from three objectives based around melodic leaps and the flow of the melody in relation to the chords. User evaluations determined that participants preferred melodies from their complete system over randomised parts of the system in terms of pleasantness, randomness and harmoniousness. MetaCompose was recently used to generate music in real-time that could express different mood-states in the Checkers game-playing environment [88].

Muńoz et al used the compositional technique of unfigured bass (creating the melody of the bass voice without specifying the upper parts) as an evolutionary compositional challenge [76]. They approached this task using a multiagent system comprised of co-adapting memes that emerged to characterise the unfigured bass technique. They notably considered memes rather than genes, where a meme was modelled as instructions that specify the procedure of a search. They used memetic search as it included a local search optimisation as part of the search strategy. Each memetic composer agent used a musical fitness function based on well-known harmony rules. The system had two steps: in the first (evolutionary) step composer agents took a bass line input and explored the space to find a suitable melody subspace; the second (local search) step completed the bass line using a set of local composer agents who exploited this subspace. Throughout this process the system employed six memetic composer agents, three evolutionary composer agents and three local composer agents. From a series of experiments they determined that their adaptive memetic algorithm was superior to conventional evolutionary music approaches, at this task.

The use of GAs for evolving Continuous Time Recurrent Neural Networks (CTRNNs) for audio synthesis was examined in the Plecto system [40]. The authors employed this as a method to use ‘low-level of abstraction’ between genotype and phenotype thus freeing the system from constraints introduced by some high-level system and hence expanding the audio search space. Plecto employed a collaborative interactive fitness measure sourced from online communities. Although this study described the musical process as audio generation rather than music composition, at a low level, the two are the same and the one process—the creation of musical sound.

In recent years a number of systems have been proposed that integrated evolutionary methods with the practice of live coding. Live coding is a practice where software that creates music, and sometimes visuals, is written and manipulated in real-time as part of a live performance [12]. The use of evolutionary techniques in generating new musical constructs during live coding has been explored in [34]. This system was built around the Extramurous platform in which performers entered Tidal code to text-boxes in a web-based client to control their musical contribution. A GA was used to evolve patterns (pieces of code) that produced music that was pleasing to the listener. The population was initialised empty and the user added individual patterns to the initial population as they encountered them. Once there were enough patterns, the population was evolved in real time using mutation with a constructed grammar and the user as the fitness measure. A framework for developing an evolutionary system that generated music using the live-coding language ChucK [100] was presented in [57]. The system proposed the development of grammars that would stochastically create valid ChucK code using GE that could then be amended by the user in realtime.

Olseng and Gambäck described a multi-objective EA that generated short MIDI melodies by developing the melody and abstract harmonization in tandem [78]. Individuals contained both a melodic and harmonic genotype which directly encoded pitches. Four fitness measures were used. The melodic local objective measures the tonality and pleasantness of the melody, the melodic global objective is based on 18 statistical features, the harmonic local objective attempts to stabilise chords vertically and the harmonic global objective judges chords in relation to other chords in the phenotype. While the system did evolve successful results, evaluations determined that the system did not compose pieces that were found to be universally appealing.

4 Discussion

An overview of the main points from some of the papers discussed in the previous section is shown in Table 1. In this table *System* is specified as either the name of the given system or the first author of the cited paper. *Problem* is the problem domain as stated in the paper. *Alg.* refers to the algorithm used. Where an algorithm is not explicitly stated in the paper, we denote it as EC. *Representation* is an indication of the type of representation used for an individual as described in the paper. *Fitness* refers to the type of fitness measure used in the system.

The most notable point from looking at an overview of this sort is to acknowledge that results can not be compared. There is no best solution, best result, or best fitness measure for the problem of 'music'.

4.1 Problem

It is evident from Table 1 that many of the problems reviewed have been merely listed as 'music composition'. Indeed it could be argued that each of these papers described the problem of music composition, yet in looking at them in detail we can see that they each posed the problem in their own way. Music composition can refer to the generation or creation of many forms of music such as polyphonic music, short melody lines or four-part harmonies/unfigured bass. These forms can be further broken down into simpler tasks. For instance where unfigured bass is named as the problem, the task at hand could be either composing the unfigured bass line itself [76], or in creating the upper voices once given a specific bass line [25]. The practicalities in the difference between such musical tasks could only be succinctly explained and understood by someone with third level knowledge of music theory. Such explanations go beyond the scope of this paper, but each problem is introduced and explained in each study. The point is that 'music composition' is a highly complicated and nuanced problem—in fact it should not be described as one specific problem but as a family of problems, each of which should be addressed in its own way. The only way to approach each individual problem is by employing sufficient amounts of domain knowledge. From this review it is clear that the authors generally have sufficient musical knowledge to embed this in each experiment. The success of any EC experiment in music composition or generation is dependent on sufficient domain knowledge being supplied and understood by the programmers of the experiment.

4.2 Algorithm

When we attempt to summarise the type of EC used in each system in Table 1, it appears that GA is the most popular type of evolutionary algorithm applied to music composition. On closer inspection however, we must acknowledge that this is somewhat of a generalisation. From examining these studies in detail we note that many authors implement their own variations of the algorithms when applying them to the problem of music. Most studies have not employed EC in a typical manner used in traditional studies. Traditional EC studies look for one best individual in the final problem. But in this musical domain many studies used the full population [3, 30, 102] or multiple individuals from the population [56] in creating the final result. Some studies didn't even measure fitness from individuals but instead examined musical trajectories across generations [30] or measured the genetic drift [23] to drive evolution. Furthermore, some studies stated that they used GAs but also included context-free grammars in their implementation thus being more indicative of a generative grammar-based or GP-like approach [49]. Many studies did not detail or discuss the implementation as the focus of their studies. DarwinTunes for instance specifies a tree-based structure which certainly implies the use of GP, although this is not stated [61]. A number of other studies referred to evolutionary systems, but again did not give explicit details as to whether they considered their system to use GA, GP or some other implementation [82, 102].

We surmise that the reasoning for this is that to the authors, the name of the algorithm used was of less importance that the augmentation of its functionality to the task at hand. In traditional GP studies on accepted benchmark problems, much importance may be placed on the details of the workings of the algorithm as it is the comparison of algorithms that is of interest. If a benchmark result can be attained, then all details of experiments must be transparent, and combinations of EC experimental criteria such as selection methods, operator probabilities, population size and number of generations must be explored and reported on. For aesthetic problems that have no benchmark, such details are less crucial. The reviewed studies, for the most part, did stipulate such attributes but the focus of these studies has been on finding an optimal method for producing music (in some form) rather than optimising any EC algorithm. Studies with such a focus do not detract from the validity of EC but rather they can only enrich the study of evolutionary methods by considering alternative or complementary methods of approaching problems.

There have also been many other successful applications of alternative population-based methods to music composition such as swarm optimisation [6] or cellular automata [73] among others, but these were not considered for this review.

4.3 Representation

The representation of an individual in an EC application to music is one of the most specific aspects of any study. Although Table 1 offers an indication of the type of representation used, full details can only be acquired by reading each paper in full. Some studies offer a description of both genotype and phenotype representation, although this is not universal across all studies. There is no standard way that a piece of music is encoded. The representation used is highly dependent on the specifics of the problem domain, but even experiments in similar domains do not necessarily use similar representations. The representation is dependent on the amount of information an individual must convey; does the individual represent a melody or harmony, does it represent pitch only or pitch and duration, are there pitch limits in place, how long is the melodic segments being represented, is translation being used through grammars, coding, MIDI interpretations? Furthermore, appropriate operators must be designed to alter such individuals. Most systems require variations of the traditional crossover and mutation or separate custom-made operators that can evolve individuals in the given representation. As in the discussion on problem domain, it is not possible to compare the merits of one representation to another; such a decision is dependent on the musical task undertaken. However, we have noted that authors of multiple variations of a system tend to use the same or highly similar representations in each of their experiments [3–5, 19, 20, 54, 56, 59]. Therefore we would advise anyone embarking on such studies to pay particular attention to the relationship between the problem domain and their representation and any limitations that each would enforce, at the beginning of any set of experiments.

4.4 Fitness measures

Once the musical problem has been succinctly described, the algorithm specified and a sensible individual representation proposed—how does one determine if a given individual is better or worse than another? This is not a trivial issue; among all the aspects of applying EC to music composition, the notion of fitness is the most difficult to consider. Studies that examine the translation of one melody to another can consider some form of distance to a target melody [38, 55, 81]. In these studies the fitness may be easily measurable, but for most compositional studies it is the merit of the resultant composition, rather than a pre-specified target, that is to be measured. Approaches to musical fitness have included interactive fitness, similarity to musical corpora, rule-based measures and the training of ANNs.

Interactive fitness is arguably the most subjective measure of fitness but it leads to the problem of the *fitness bottleneck* due to the cost in time and money to implement such measures [3]. Furthermore, such experiments will naturally lean towards the preferences of the person giving the fitness judgements and hence cannot be considered objective. More recently crowd based collaborative fitness measures from online participants have been used [40, 61]. Although these are based on human perception, they are not truly interactive as the judgements are not generally used as fitness in real-time. Furthermore, while such online crowd-based measures are cheaper and more easy to access, they are likely to be heavily biased in age, gender and experience depending on the sourcing of volunteers. A number of studies have used ANNs trained on previous human judgements as fitness measures. While this may seem like an efficient way to replace a human observer and address the fitness bottleneck, a number of authors from such papers have concluded that these measures were lacking in comparison to human judgments [5, 44].

Fitness measures based on a corpus of music or a set of musical rules can again suffer from biases. Experiments that use a musical corpus to derive a genre-based fitness measure will always be limited to creating music in a similar style to the given corpus thus limiting the creative potential of the system. A number of systems employed fitness functions that were based on musical measures, either through harmonic rules or statistical measures. For such a fitness function to indicate true musical creativity, however, they would need to be heavily justified. Such measures drive the evolution, but regardless as to how involved or multi-faceted a given fitness measure is, it is surely impossible to say that this is the ultimate measure of music. Generally, the fitness measure is derived as part of the experiment; the fitness drives evolution in some meaningful manner through the search space. While each experiment justifies its choice in fitness measure, there is no one particular 'best' measure by which to select good music over bad.

One other measure that has been used in musical systems but would not be normally seen in EC experiments is the idea of random fitness. Random fitness is generally only employed when the full population is used as part of the result [3, 23, 102]. The notion of random fitness is alien to EC researchers as it is the fitness measure that drives evolution, and a random measure cannot achieve meaningful search. However, in systems that examine the process of evolution or consider the manner in which individuals combine within a population to form a single composition, this notion of searching towards an objective becomes less explicit. This use of random, obscure or implicit measures is one of the most interesting amendments to systems that examine music and other artistic applications.

A review of fitness measures used in evolutionary art and music in the early years of EvoMUSART was presented by Johnson [45]. At a similar time a separate overview of the most prevalent measures and ideas used to examine and evaluate melodies was given by De Freitas [23]. They discussed ten attributes used in the evaluation of melodies based on pitch and rhythm measurements, concluding that previous approaches to formalise a fitness function for melodies have not comprehensively incorporated all measures. It does not appear that there is a simple solution to the creation of a musical fitness function. However, one option that is of particular interest is that of an intrinsic fitness measure, a measure of performance not stated from the outset but emergent from the system itself. Alternative external measures such as the progressions of genetic drift in [23] or the internal self-circular creation of critics and melodies in [59] could lead to more interesting systems that do not merely follow arbitrarily created rules for the given experiment but emulate complex self-adaptive systems that could explore a search-space through implicitly generated ideals.

The interplay of domain, representation and fitness will determine what a given system can and will produce: the search domain determines the space (or 'style') of music, the representation limits the individuals that can be created within that space and the fitness measure determines which individuals will be considered a good result. In EC experiments we know that fitness drives the search. However, it is a mistake to assume that the fitness measures in an EC music system drives towards musicality. Yes, we hope to emulate musicality in such systems, but if musicality does not lie in the fitness function it could instead lie in the domain, representation, initial population. In such cases the search is not *towards* musicality, the system is optimising towards another objective in a musical domain. But as creativity (or in this case musicality) is such a difficult concept to define, as discussed in Sect. 5, an indirect search with high levels of exploration may be the best method to attempt to find it.

4.5 Evaluation

Some studies consider music to be a stagnant object, others consider it a process; it has even been argued that music does not exist [104]. Let us assume for the purpose of this discussion that it does exist and that, given appropriate context, either this music or the process that created it can be measured in some way.[3] A lack of evaluation of creative systems has been noted many times in recent years [7, 47]. Evaluation on autonomous systems is required to determine if they are performing as expected. However, as with the problem of fitness, evaluating the performance of a music generation EC system is not as straight-forward as evaluation on more traditional problems. Fitness plots can be examined—but even if an average vs. best fitness plot over 50 generations averaged over 40 runs showed a typical fitness improvement—does this mean that the system succeeded in creating good music? If such results were seen we could certainly say that the evolution has occurred as expected; if our problem domain was well defined we could confidently state that it created music; if our individual representation was sufficiently constrained we may even speculate that it created good music. But—can we determine if the evolutionary process has autonomously created music over successive generations? This is a more complicated question to address.

When discussing a compositional system, the temptation is to evaluate the system on the melodic output, often through performing human listening tests. What constitutes ‘good music’ is subjective to the individual however. Human listening tests may not result in objective or reproducible results. Furthermore, performing evaluations purely on the output of a system can limit the validity of your evaluation. The output of a musical system may be interesting, but if it is the system’s ability to compose being investigated, it is imperative to look beyond the generated output. The relevance of this distinction is dependent on the focus of one’s research. Pearce et al specified two ways in which machine composers may be evaluated: in terms of the music they composed and the manner in which they composed [79]. There are many generative music systems whose purpose was to compose music while other studies were more focussed on the academic exploration of autonomous musicality or creativity. Music systems focussed on ‘mere generation’ were defended in [29], in highlighting that much music innovation has been achieved in Musical Metacreation (MuMe [77]) from generative systems focussed on human-interactive co-creativity. Any music compositional system fits somewhere on a spectrum between pure generation (the musical output is most important) and pure computational creativity (the behaviour of the system is most important). Meaningful and relevant evaluation of any system is dependent on where the system lies within such a spectrum.

In the context of applying EC methods to music composition it is hence important not to automatically limit evaluations to those taken based on human measurements from the produced musical output. Limitations that can be incurred by such measure have been discussed at length by Loughran and O’Neill [60].

[3] Wiggins, who made this argument, has authored many papers on EC applied to music and we hope will forgive this assumption in this context.

4.6 Outstanding issues

In 2005 Jon McCormack proposed five Open Problems in Evolutionary Music and Art [68], revised in [67]. These problems considered representations and mechanisms that were both robust to modifications, effective and suitable fitness functions, systems that evolved artefacts recognisable as *art*, systems where agents created and recognised their own creativity and the development of an 'art theory' of such systems. Each of these raised an important issue in EC applied to any creative process including representation, fitness, the purpose of such systems, self-referential (or even self-aware) systems and what we may learn from such systems. The author was correct in concluding that such grand challenges would not be solved in the immediate future; these questions remain open today. Each of these problems deserve much discussion, but the one we would like to draw attention to is Problem number three. This proposes:

> To create evolutionary music and art systems that produce art recognized by humans for its *artistic* contribution (as opposed to any purely technical fetish or fascination).

While we agree that, for most systems, the output should be of interest, this should not exclude systems whose contribution lies within the system rather than the generated output. As with our argument above regarding limitations from evaluation, it is imperative that we look past traditional aesthetic evaluations in determining the merit of autonomous music systems. McCormack does address this point in proposing the idea of 'art-as-it-could-be' and the idea of a wider movement in machine generated art. Furthermore later problems in the paper were more focussed on the operations of the system and what could possibly be learned from a developed system, indicating that the purpose of such systems can be found in other aspects apart from what they produce. In a sense, if Problem number four (self-awareness of creativity) and Problem number five (art theory) were to be solved, Problem number three may not be such a problem anymore.

In an early discussion on EC applied to AC, Wiggins et al determined that GAs could be useful in musical tasks but only on small constrained tasks [103]. They argued that because evolutionary algorithms are heuristic search methods and lack structure in their reasoning they will never be able to replicate or simulate the human thought behaviour undertaken in music composition. While this may be true, this does not mean that EC cannot be useful in examining the creative act of music composition. The idea of replacing the full compositional process with one autonomous system may not be possible, but as we have seen from this review, most systems tackle a small domain-constrained aspect of composition. Breaking the act of composition into modules is not unlike the human-composer approach; compositions are not created instantly but rather through a series of iterations. While no one autonomous system could replace a composer, it is possible that they could replace part of the composition process. In time, might some hierarchical evolution of evolution structure be able to theoretically approach the whole problem? We propose that systems should be developed that remain within sensible constraints, while being applied and exposed to increasingly creative challenges.

Tatar and Pasquier recently published a review of all autonomous agents that have been applied to the problem of music composition. They stated towards the end of this paper that they had 'not found any musical agents applying GP in their system design.' While they may not have met the authors' definition of 'musical agent' we have noted several studies that have directly applied GP[4] to some aspect of algorithmic composition [19, 20, 36, 44, 64, 81, 92, 93, 95]. This is not to negate the validity of Tatar and Pasquier's comprehensive review. Instead we wish to highlight that in reviews of the field of AC, GP systems are not making enough impact to be found. This is something that the community should aim to rectify.

5 EC and the search for creativity

The emerging field of CC has been growing in momentum as a recognised and valued sub-field of AI for a number of years [18]. Those that would once have assumed and off-handedly remarked such comments as 'Obviously computers can never be creative' may have to re-consider their words. Such a derisive attitude is not new however. As far back as 1982 Marvin Minsky argued that making such statements bordered on silly, and were mainly due to the fact that people considered creativity to be a kind of super-ability afforded only to the few geniuses among us [72]. As he stated in his essay:

> There's a big difference between "impossible" and "hard to imagine." The first is about *it*; the second is about *you*!

Instead he proposed that there was no substantial difference between ordinary thought and creative thought and that as such there is no such thing as 'creativity' at all. This sentiment is similar although the conclusion starkly different from Boden's argument whereby she stated that creativity is merely an aspect of human intelligence. Whether you follow Minksy's or Boden's conclusion as to the existence of creativity, they are in agreement that creative thought or creative ability is merely part of general intelligence and not an abstract phenomenon requiring specialised explanation. It follows that if AI has become so important in modern computer science and creativity is a part of general intelligence, then the study of CC must also be considered a matter of importance.

Much of the confusion or negativity in relation to the idea of computers being creative stems from a lack of definition and understanding as to what creativity actually means. While people may believe they have an understanding of what creativity means, a technical definition still alludes us. The standard definition of creativity is succinct [87]:

> Creativity requires both originality and effectiveness

[4] And many more that used GE, see Table 1.

Unfortunately this is far from the only definition used. It has been stated that there are over a hundred definitions in the literature [71], but considering the subject is studied in so many varied fields and is now entering computational fields, this number is likely to be much higher than this. This lack of definition leads to difficulties in measuring or evaluating creativity—as how can we measure what we cannot define?

It is generally accepted that for a system to be deemed creative it must display novelty and value. There are two distinct variations to the term ‘novel’. Ideas that are novel to the individual are considered Psychologically (P) Creative, whereas ideas that are novel to the world, that have not been derived before, are said to be Historically (H) Creative [9]. This confusion between H and P creativity can attribute to the colloquial notion that creativity must involve big important creative acts or achievements. In fact it is P-creativity that is of interest to computer programmers; from a computational stance it does not matter if the world has seen this creative act—as long as it is novel to the system under development. The concept of ‘value’ is dependent on the purpose of the system under development. More recently, some researchers have added that for a system to be deemed creative it must display novelty, value and intent [99]. This third criteria ‘intent’ is likely the most difficult criteria to demonstrate. When a human creates something, it can be assumed they did so with intent, but when a computer system creates, intent must be explicitly determined.

Evaluating whether or not a system is creative is hindered by the lack of definition of creativity and the lack of consensus as to what should be measured to determine creativity. The lack of evaluation in CC systems has been noted throughout the development of the field [7, 15, 47]. This issue has been addressed in recent years and a number of evaluative frameworks have been proposed including the Lovelace Test [11], the Creative Tripod [17], Turing-style tests [1] and the Standardised Procedure for Evaluating Creative Systems (SPECS) [48]. We noted earlier in this paper that music generative systems can be purposed along a scale from mere generation to genuine creativity. Systems that focus on automating part of the compositional process can be considered generative, those that augment the human compositional experience are co-creative and those that focus on emulating creativity either through generating an artefact or in understanding the process are aiming to be truly computationally creative. Thus we can use evolutionary (and indeed any computational) systems both in generating creative music or in the search for the understanding of creativity. Ideally, we would like to be able to determine the scale or level of the creativity that is obtainable from such systems; we could consider the experiments listed in Table 1 along such a scale. The difficulties in evaluating creativity, however, make such comparisons extremely difficult and we would not wish misrepresent any work by classifying it in such a way. Nevertheless, we would advise future creators of generative systems to bear such considerations in mind when evaluating their systems and presenting results.

Boden has stated that there are three types of creativity that may be observed: combinational, explorational and transformational [8]. Combinational creativity combines familiar ideas resulting in a new idea or concept. Explorational creativity searches the ‘conceptual space’—a space defined and constrained by the domain under consideration. Transformational creativity results in a transformation of the conceptual space itself. Boden regarded transformational as the type of creativity with the greatest opportunity for discovery, although it may be the most difficult to evaluate. The processes of exploration and combination are reminiscent of the processes involved in EC; mutation operators explore the search space and crossover combines individuals in the space for new ideas. Also, the use of grammars or genotype-phenotype mappings in GP systems offer a platform for the transformation of ideas and concepts. Furthermore, population based search heuristics such as EC begin searching from multiple start points allowing much better exploitation and coverage of the conceptual space than single point search heuristics. As we have noted, in EC systems there is a variety of ways in which to define fitness—including methods that consider the full population or the relationship between individuals in the population. Hence there is more scope and possibility to search towards less specified and more relative ideas, such as creativity, than there may be with other error-based ML methods. It is thus unsurprising that so many systems are emerging that apply, and as we have seen amend, EC algorithms for the purpose of creative studies.

In the mid 1990s Spector and Alpern [92] noted the challenge of considering cultural relevance in rigorously adjudicating the output of an artistic system. They argued against evolving aesthetically towards rules derived from a given form or genre on three grounds: that this genre may be only formalisable because it is currently dead, that adherence to rules of a form does not necessarily indicate aesthetic value and that it is not clear if such values or rules would generalise to other genres. Instead they proposed that in using EC methods they could factor out aesthetic judgement and instead develop a set of critical criteria which the system was capable of conforming to. By providing a way to separate aesthetic judgement from system judgements they proposed that EC methods could offer a better opportunity to explore creative space:

> The new technologies of GAs and GP offer the promise of tractable evolutionary processing, and hence theories of creativity-through-evolution may now be explored experimentally.

In an early discussion on algorithmic composition as a model of creativity, Jacob stated that there were two types of creativity: genius and hard work [43]. In this he proposed that the first type consisted of inspirational ideas that could not be explained and therefore could not be re-produced. His second type of creativity was more akin to incremental revisions which could be replicated algorithmically in an iterative manner. We do not agree with this view of creativity but rather view that elements of both of these 'types' are needed for the emergence of creativity. This notion of unexplainable 'genius' is akin to the fallible argument of specialised creativity argued against by Minsky and Boden. But iterative search on its own is also not likely (albeit possible) to result in creativity. Unexplainable inspiration is not necessary, but search is not necessarily sufficient for creativity to be present.

From these discussions around the meaning and pursuit of creativity, it is evident that creativity can only be attained through exploration and search. In the Blind Watchmaker, Richard Dawkins stated [21]:

> Effective searching procedures become, when the search space is *sufficiently* large, indistinguishable from true creativity.

We do not wholly agree with this statement however, particularly with the emphasis on 'sufficiently'. If the emphasis instead was to be place on *effective* we believe the statement may carry more weight. The size of the search space is irrelevant to the possibility of creativity. Imagine if we could consider the search space to encompass the real world. In this case it must contain creativity, as we know it. Searching through this space is not creativity, but searching it effectively could lead to creativity. It is this idea of effective search that is imperative for the emergence of creativity from heuristic search methods such as EC.

5.1 EC for music composition

In conducting this review we noted that quite a number of studies made statements such as 'other studies only considered…' before proceeding to state that the new idea considered in their study was the one that enabled it to autonomously create music. The problem with such arguments is that no individual system could possibly, at this time, autonomously compose music. To claim that your system 'solves' the problem is doomed to be simply wrong. We have noted many variations within the problem domain, representation, fitness measures, algorithmic implementation and evaluation methods. There is no one combination of these requirements that will outperform all other systems. We should not be designing EC systems to be able to produce the 'best music'. However, neither should such systems be focussed on composing music to test for the best architecture or experimental set-up within EC. Without a measurable objective as to what constitutes a 'good' result we cannot determine a definitive answer as to whether GAs, GP, GE or some other variation is better at composing music let alone if a particular operator or selection method is superior to others; the question is non-sensical.

But in the scientific world we must aspire to create systems that are improvements on what has come before. If these systems are not to generate the best music and not to improve EC functionality, then what is their purpose? We believe that the answer to this lies in the above discussion on creativity. The search for creativity relies on effective exploration of the search space. In EC experiments we can limit the search space through domain constraints and individual representations, we can direct the search implicitly or explicitly through the design of fitness measures, we can transform individuals and domains through grammars, mappings and operators. In EC experiments, we start with a population of solutions, not an individual. As seen from the studies reviewed we can use the whole population, an individual result, a few individuals from the population or consider how the population changes within the process of composition. Evolutionary methods allow us to constrict or expand the data and processes necessary for exploratory creative search. Such methods have proven to be highly versatile in their approach to algorithmic music composition. We hope that this versatility will grow and be encouraged in future studies.

6 Conclusion

We have presented a review on the application of EC methods to the creative task of algorithmic music composition. With the growth of events such as EvoMUSART, the number of papers encompassing both of these fields has steadily increased over the last three decades. We detailed the problem domains, methods, representations and fitness measures used in numerous studies from the early 1990s to today. We noted throughout this paper that the application domain of *music composition* can refer to a large number of musical sub-tasks which must be clearly defined through domain and representation constraints. Even when such constraints are in place, one of the biggest challenges for aesthetic applications such as music, in comparison to more traditional problems such as symbolic regression, is that there is no simple way to measure the 'fitness' of a piece of music. While fitness measures have been developed that follow rules, mimic given corpora or rely on human judgment, we noted that no single fitness measure can autonomously, objectively and reliably determine what is good music; *musicality* is not an easily defined objective.

We introduced the field of CC and examined the nature of creativity and its relation to evolutionary methods. We addressed some of the modern day objections to the notion of computers being creative. As noted from the opening of this paper, music and computers have witnessed a long history of complementing each other. The simulation of creative acts such as music composition and the ability of a computer program to genuinely display creativity is a philosophical question as well as a technical challenge. Music, among the other creative arts, poses a different challenge than numerical tasks in that there is no correct answer. It has been claimed that music does not exist [104] and it has been claimed that creativity does not exist [72]. But if such things did not exist, there would be no way to even attempt to measure or emulate them computationally. Instead of dwelling on such arguments, we propose to continue the exploration of musical tasks and challenges using the versatility of GP and evolutionary methods in the pursuit of identifying, examining and ideally exhibiting musical creativity.

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. C. Ariza, The interrogator as critic: the turing test and the evaluation of generative music systems. Comput. Music J. **33**(2), 48–70 (2009)
2. Association for computational creativity. http://computationalcreativity.net/
3. J. Biles, GenJam: a genetic algorithm for generating jazz solos, in *Proceedings of the International Computer Music Conference* (International Computer Music Association, 1994), pp. 131–131
4. J.A. Biles, Straight-ahead jazz with GenJam: a quick demonstration, in *MUME 2013 Workshop* (2013)
5. J. Biles, P. Anderson, L. Loggi, Neural network fitness functions for a musical iga. Accessed from https://scholarworks.rit.edu/other/184 (1996)

6. T. Blackwell, J. Branke, Multi-swarm optimization in dynamic environments, in *Workshops on Applications of Evolutionary Computation* (Springer, 2004), pp. 489–500
7. M.A. Boden, Creativity and artificial intelligence. Artif. Intel. **103**(1), 347–356 (1998)
8. M.A. Boden, *The Creative Mind: Myths and Mechanisms* (Psychology Press, London, 2004)
9. M.A. Boden, Computer models of creativity. AI Mag. **30**(3), 23 (2009)
10. A. Brabazon, M. O'Neill, S. McGarraghy, *Natural Computing Algorithms* (Springer, Berlin, 2015)
11. S. Bringsjord, P. Bello, D. Ferrucci, Creativity, the turing test, and the (better) lovelace test, in *The Turing Test*, ed. by J.H. Moor (Springer, Berlin, 2003), pp. 215–239
12. A.R. Brown, A. Sorensen, Interacting with generative music through live coding. Contemp. Music Rev. **28**(1), 17–29 (2009)
13. A.R. Burton, T. Vladimirova, Genetic algorithm utilising neural network fitness evaluation for musical composition, in *Artificial Neural Nets and Genetic Algorithms*, ed. by G.D. Smith, et al. (Springer, Berlin, 1998), pp. 219–223
14. A.R. Burton, T. Vladimirova, Generation of musical sequences with genetic techniques. Comput. Music J. **23**(4), 59–73 (1999)
15. A. Cardoso, T. Veale, G.A. Wiggins, Converging on the divergent: the history (and future) of the international joint workshops in computational creativity. AI Mag. **30**(3), 15 (2009)
16. C. Chafe, *Glitch Free FM Vocal Synthesis. Center for Computer Research in Music and Acoustics* (Stanford University, Stanford, 2013)
17. S. Colton, Creativity versus the perception of creativity in computational systems, in *AAAI Spring Symposium: Creative Intelligent Systems* (2008), pp. 14–20
18. S. Colton, G.A. Wiggins et al., Computational creativity: The final frontier? ECAI **12**, 21–26 (2012)
19. P. Dahlstedt, Autonomous evolution of complete piano pieces and performances, in *Proceedings of Music AL Workshop* (Citeseer, 2007)
20. P. Dahlstedt, *Sounds Unheard of: Evolutionary Algorithms as Creative Tools for the Contemporary Composer* (Chalmers University of Technology, Goteborg, 2004)
21. R. Dawkins, The blind watchmaker: why the evidence of evolution reveals a universe without design (reissued in 2006 and 1996) (1996)
22. A.R. de Freitas, F.G. Guimarães, Originality and diversity in the artificial evolution of melodies, in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (ACM, 2011), pp. 419–426
23. A.R. de Freitas, F.G. Guimaraes, R.V. Barbosa, Ideas in automatic evaluation methods for melodies in algorithmic composition, in *Sound and Music Computing Conference* (2012)
24. A.O. de la Puente, R.S. Alfonso, M.A. Moreno, Automatic composition of music by means of grammatical evolution. in *ACM SIGAPL APL Quote Quad* vol. 32 (ACM, 2002), pp. 148–155
25. R. De Prisco, G. Zaccagnino, R. Zaccagnino, Evobasscomposer: a multi-objective genetic algorithm for 4-voice compositions, in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (ACM, 2010), pp. 817–818
26. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
27. G. Diaz-Jerez, Composing with melomics: delving into the computational world for musical inspiration. Leonardo Music J. **21**, 13–14 (2011)
28. P. Donnelly, J. Sheppard, Evolving four-part harmony using genetic algorithms, in *European Conference on the Applications of Evolutionary Computation* (Springer, Berlin, Heidelberg, 2011), pp. 273–282
29. A. Eigenfeldt, O. Bown, A.R. Brown, T. Gifford, Flexible generation of musical form: beyond mere generation, in *Proceedings of the Seventh International Conference on Computational Creativity* (2016)
30. A. Eigenfeldt, P. Pasquier, Populations of populations: composing with multiple evolutionary algorithms, in *Evolutionary and Biologically Inspired Music, Sound, Art and Design* (Springer, Berlin, 2012), pp. 72–83
31. A. Gartland-Jones, Musicblox: A real-time algorithmic composition system incorporating a distributed interactive genetic algorithm, in *Workshops on Applications of Evolutionary Computation* (Springer, 2003), pp. 490–501
32. A. Gartland-Jones, P. Copley, The suitability of genetic algorithms for musical composition. Contemp. Music Rev. **22**(3), 43–55 (2003)

33. P. Gibson, J. Byrne, Neurogen, musical composition using genetic algorithms and cooperating neural networks, in *Second International Conference on Artificial Neural Networks* (IET, 1991), pp. 309–313
34. S. Hickinbotham, S. Stepney, Augmenting live coding with evolved patterns, in *International Conference on Evolutionary and Biologically Inspired Music and Art* (Springer, 2016), pp. 31–46
35. L.A. Hiller Jr, L.M. Isaacson, Musical composition with a high speed digital computer, in *Audio Engineering Society Convention*, vol 9 (Audio Engineering Society, 1957)
36. D.M. Hofmann, A genetic programming approach to generating musical compositions, in *International Conference on Evolutionary and Biologically Inspired Music and Art* (Springer, 2015), pp. 89–100
37. A.K. Hoover, M.P. Rosario, K.O. Stanley, Scaffolding for interactively evolving novel drum tracks for existing songs, in *Workshops on Applications of Evolutionary Computation* (Springer, 2008), pp. 412–422
38. A. Horner, D.E. Goldberg, Genetic algorithms and computer-assisted music composition. Urbana **51**(61801), 437–441 (1991)
39. D. Horowitz, Generating rhythms with genetic algorithms, in *AAAI* vol 94 (Cambridge, 1994), p. 1459
40. S. Ianigro, O. Bown, Plecto: a low-level interactive genetic algorithm for the evolution of audio, in *International Conference on Evolutionary and Biologically Inspired Music and Art* (Springer, 2016), pp. 63–78
41. International conference on computational intelligence in music, sounds, art and design (EvoMUSART). http://www.evostar.org/2019/
42. B. Jacob, Composing with genetic algorithms, in *International Computer Music Conference (ICMC), Banff Alberta* (International Computer Music Association, 1995)
43. B.L. Jacob, Algorithmic composition as a model of creativity. Organ. Sound **1**(3), 157–165 (1996)
44. B. Johanson, R. Poli, *GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters* (Citeseer, Princeton, 1998)
45. C.G. Johnson, Fitness in evolutionary art and music: what has been used and what could be used?, in *Evolutionary and Biologically Inspired Music, Sound, Art and Design* (Springer, Berlin, Heidelberg, 2012), pp. 129–140
46. C.G. Johnson, J.J.R. Cardalda, Genetic algorithms in visual art and music. Leonardo **35**(2), 175–184 (2002)
47. A. Jordanous, Evaluating evaluation: assessing progress in computational creativity research (2011)
48. A. Jordanous, A standardised procedure for evaluating creative systems: computational creativity evaluation based on what it is to be creative. Cognit. Comput. **4**(3), 246–279 (2012)
49. Y. Khalifa, B.K. Khan, J. Begovic, A. Wisdom, A.M. Wheeler, Evolutionary music composer integrating formal grammar, in *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation* (ACM, 2007), pp. 2519–2526
50. S.O. Kimbrough, G.J. Koehler, M. Lu, D.H. Wood, On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: distance tracing and no free lunch. Eur. J. Oper. Res. **190**(2), 310–327 (2008)
51. K. Kunimatsu, Y. Ishikawa, M. Takata, K. Joe, A music composition model with genetic programming-a case study of chord progression and bassline, in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2015), p. 256
52. P. Laine, M. Kuuskankare, Genetic algorithms in musical style oriented generation, in *Evolutionary Computation, 1994. Proceedings of the First IEEE Conference on IEEE World Congress on Computational Intelligence* (IEEE, 1994), pp. 858–862
53. Library of Congress, 2009 national recording registry (2009) https://www.loc.gov/item/prn-10-116/
54. R. Loughran, J. McDermott, M. O'Neill, Grammatical evolution with zipf's law based fitness for melodic composition, in *Sound and Music Computing Conference*, Maynooth (2015)
55. R. Loughran, J. McDermott, M. O'Neill, Grammatical music composition with dissimilarity driven hill climbing, in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. ed. by J. Correia, C. Johnson, V. Ciesielski (Springer, Berlin, 2016), pp. 110–125

56. R. Loughran, J. McDermott, M. O'Neill, Tonality driven piano compositions with grammatical evolution, in *2015 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2015), pp. 2168–2175
57. R. Loughran, M. O'Neill, 'My Little Chucky': towards live-coding with grammatical evolution, in *Musical Metacreation (MuMe)* ed. by P. Pasquier, O. Bown, A. Eigenfeldt (2017)
58. R. Loughran, M. O'Neill, Clustering agents for the evolution of autonomous musical fitness, in *Evolutionary and Biologically Inspired Music, Sound, Art and Design* (Springer, Berlin, 2017)
59. R. Loughran, M. O'Neill, The popular critic: evolving melodies with popularity driven fitness, in *Musical Metacreation (MuMe)*, Paris (2016)
60. R. Loughran, M. O'Neill, Limitations from assumptions in generative music evaluation. J. Creat. Music Syst. **2**, 1 (2017)
61. R.M. MacCallum, M. Mauch, A. Burt, A.M. Leroi, Evolution of music by public choice. Proc. Natl. Acad. Sci. **109**(30), 12081–12086 (2012)
62. P. Machado, A. Cardoso, Nevar-the assessment of an evolutionary art tool, in *Proceedings of the AISB00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, vol 456 (Birmingham, UK, 2000)
63. P. Machado, J. Tavares, A. Cardoso, F.B. Pereira, E. Costa, Evolving creativity, in *First International Joint Workshop on Computational Creativity* (Universidad Complutense de Madrid, Madrid, 2004), pp. 17–19
64. B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, J. Romero, A corpus-based hybrid approach to music analysis and composition, in *Proceedings of the National Conference on Artificial Intelligence*, vol 22 (MIT Press, Cambridge, 2007), p. 839
65. B. Manaris, D. Vaughan, C. Wagner, J. Romero, R.B. Davis, Evolutionary music and the Zipf–Mandelbrot law: developing fitness functions for pleasant music, in *Applications of Evolutionary Computing* (Springer, Berlin, 2003), pp. 522–534
66. B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, R.B. Davis, Zipf's law, music classification, and aesthetics. Comput. Music J. **29**(1), 55–69 (2005)
67. J. McCormack, Facing the future: evolutionary possibilities for human–machine creativity, in *The Art of Artificial Evolution,* ed. by J.J. Romero (Springer, Berlin, 2008), pp. 417–451
68. J. McCormack, Open problems in evolutionary music and art, in *Applications of evolutionary computing*. (Springer, Berlin, 2005), pp. 428–436
69. J. McDermott, U.M. O'Reilly, An executable graph representation for evolutionary generative music, in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (ACM, New York, USA, 2011), pp. 403–410
70. L.F. Menabrea, A. Lovelace, Sketch of the analytical engine invented by charles babbage (1842)
71. P. Meusburger, Milieus of creativity: the role of places, environments, and spatial contexts, in *Milieus of Creativity* (Springer, Berlin, 2009), pp. 97–153
72. M.L. Minsky, Why people think computers can't. AI Mag. **3**(4), 3 (1982)
73. E.R. Miranda, On the evolution of music in a society of self-taught digital creatures. Digit. Creat. **14**(1), 29–42 (2003)
74. E.R. Miranda, J. Al Biles, *Evolutionary Computer Music* (Springer, Berlin, 2007)
75. A. Moroni, J. Manzolli, F. Von Zuben, R. Gudwin, Vox populi: an interactive evolutionary system for algorithmic music composition. Leonardo Music J. **10**, 49–54 (2000)
76. E. Munoz, J. Cadenas, Y.S. Ong, G. Acampora, Memetic music composition. IEEE Trans. Evol. Comput. **20**(1), 1 (2016)
77. Musical Metacreation (MuMe). http://musicalmetacreation.org
78. O. Olseng, B. Gambäck, Co-evolving melodies and harmonization in evolutionary music composition, in *International Conference on Computational Intelligence in Music, Sound, Art and Design* (Springer, 2018), pp. 239–255
79. M. Pearce, G. Wiggins, Towards a framework for the evaluation of machine compositions, in *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences* (Citeseer, 2001), pp. 22–32
80. S. Phon-Amnuaisuk, E.H.H. Law, H.C. Kuan, Evolving music generation with som-fitness genetic programming, in *Workshops on Applications of Evolutionary Computation* (Springer, 2007), pp. 557–566
81. A. Pirnia, J. McCormack, Compressed multidimensional trees for evolutionary music representation, in *ICMC* (2012)

82. C.S. Quintana, F.M. Arcas, D.A. Molina, J.D.F. Rodriguez, F.J. Vico, Melomics: a case-study of ai in spain. AI Mag. **34**(3), 99–103 (2013)
83. J. Reddin, J. McDermott, M. O'Neill, Elevated pitch: automated grammatical evolution of short compositions, in *Applications of Evolutionary Computing* ed. by M. Giacobini, et al. (Springer, 2009), pp. 579–584
84. J. Reichardt, J. Reichardt, *Cybernetics, art and ideas* (Studio Vista, London, 1971)
85. J. Romero, P. Machado, A. Carballal, Guest editorial: special issue on biologically inspired music, sound, art and design. Genet Program Evolvable Mach **14**, 281 (2013). https://doi.org/10.1007/s10710-013-9191-z
86. J.J. Romero, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music* (Springer, Berlin, 2008)
87. M.A. Runco, G.J. Jaeger, The standard definition of creativity. Creat. Res. J. **24**(1), 92–96 (2012)
88. M. Scirea, P. Eklund, J. Togelius, S. Risi, Evolving in-game mood-expressive music with metacompose, in *Audio Mostky 2018: Sound in Immerision and Emotion (AM'18)*, New York (2018)
89. M. Scirea, J. Togelius, P. Eklund, S. Risi, Metacompose: A compositional evolutionary music composer, in *International Conference on Evolutionary and Biologically Inspired Music and Art* (Springer, 2016), pp. 202–217
90. J. Shao, J. McDermott, O'Neill, M., A. Brabazon, Jive: A generative, interactive, virtual, evolutionary music system, in *Applications of Evolutionary Computation*, ed. by C. Di Chio et el. (Springer, Berlin, 2010), pp. 341–350
91. K. Sims, Interactive evolution of equations for procedural models. Vis. Comput. **9**(8), 466–476 (1993)
92. L. Spector, A. Alpern, Criticism, culture, and the automatic generation of artworks, in *AAAI* (1994), pp. 3–8
93. L. Spector, A. Alpern, Induction and recapitulation of deep musical structure. Proc. Int. Jt. Conf. Artif. Intell. **95**, 20–25 (1995)
94. K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies. Evol. Comput. **10**(2), 99–127 (2002)
95. C. Sulyok, A. McPherson, C. Harte, Corpus-taught evolutionary music composition, in *Proceedings of the 13th European Conference on Artificial life, York, UK* (2015), pp. 587–594
96. K. Thywissen, GeNotator: an environment for exploring the application of evolutionary techniques in computer-assisted composition. Organ. Sound **4**(02), 127–133 (1999)
97. C.K. Ting, C.L. Wu, C.H. Liu, A novel automatic composition system using evolutionary algorithm and phrase imitation. IEEE Syst. J. **11**, 1284 (2015)
98. P.M. Todd, G.M. Werner, Frankensteinian methods for evolutionary music composition, in *Musical Networks: Parallel Distributed Perception and Performace*, ed. by N. Griffith, P.M. Todd (MIT Press, Cambridge, 1999)
99. D. Ventura, How to build a CC system, in *Proceedings of the 8th International Conference on Computational Creativity, Atlanta* (2017)
100. G. Wang, P.R. Cook, et al., Chuck: a concurrent, on-the-fly, audio programming language, in *ICMC* (2003)
101. R. Waschka, Theories of evolutionary algorithms and a`new simplicity'opera: making sappho's breath, in *Artificial Life Models for Musical Applications* (Cosenza, 2011), pp. 79–86
102. R. Waschka II, Composing with genetic algorithms: GenDash, in *Evolutionary Computer Music*, ed. by E.R. Miranda, J.A. Biles (Springer, Berlin, 2007), pp. 117–136
103. G. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, A. Tuson, Evolutionary methods for musical composition. Dai Research Paper (1998)
104. G.A. Wiggins, D. Müllensiefen, M.T. Pearce, On the non-existence of music: why music theory is a figment of the imagination. Musicae Scientiae **14**(1 suppl), 231–255 (2010)
105. G.K. Zipf, *Human behavior and the principle of least effort* (Addison-Wesley Press, Boston, 1949)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.