Noname manuscript No.
(will be inserted by the editor)

Affective Evolutionary Music Composition with MetaCompose

Marco Scirea · Julian Togelius · Peter Eklund · Sebastian Risi

Received: date / Accepted: date

Abstract This paper describes the MetaCompose music generator, a compositional, extensible framework for *affective* music composition. In this context 'affective' refers to the music generator's ability to express emotional information. The main purpose of MetaCompose is to create music in real-time that can express different mood-states, which we achieve through a unique combination of a graph traversal-based chord sequence generator, a search-based melody generator, a pattern-based accompaniment generator, and a theory for mood expression. Melody generation uses a novel evolutionary technique combining FI-2POP with multi-objective optimization. This allows us to explore a Pareto front of diverse solutions that are creatively equivalent under the terms of a multi-criteria objective function. Two quantitative user studies were performed to evaluate the system: one focusing on the music generation technique, and the other that explores valence expression, via the introduction of dissonances. The results of these studies demonstrate (i) that each part of the generation system improves the perceived quality of the music produced, and (ii) how valence expression via dissonance produces the perceived affective state. This system, which can reliably generate affect-expressive music, can subsequently be integrated in any kind of interactive application (e.g. games) to create an adaptive and dynamic soundtrack.

Keywords Evolutionary computing, genetic algorithm, music generation, affective music, creative computing

Contents

1 Introduction 2
2 Background 3

M. Scirea, P. Eklund and S.Risi
IT University of Copenhagen, Denmark
E-mail: msci@itu.dk
WWW homepage: marcoscirea.com

Julian Togelius
New York University, USA

3 Methods . . . 7
4 MetaCompose . . . 8
5 Composition Generation . . . 10
6 Real-Time Affective Music Composer . . . 16
7 MetaCompose Archive . . . 18
8 Evaluation of the Composition Generation . . . 19
9 Evaluation of valence expression through dissonance . . . 24
10 Conclusion and future work . . . 27

1 Introduction

The are many reasons to build computer generated music systems, including adapting to a dynamic environment, performing concurrently with a human player, reflecting upon music composition practice, and others [70]. Music has the power to evoke moods and emotions – even music generated algorithmically [47]. Thus music is able to enhance the experience of related media. In some cases the main purpose of a music generation algorithm is to evoke a particular mood. This is true for generators that form part of interactive systems, such as those supporting computer games. In these systems, a common goal of music generation is to elicit a particular mood that dynamically suits the current state of the game-play. Music generation for computer games can be seen as an instance of the experience-driven procedural content generation framework (*EDPCG*) [96], where the game adaptation mechanism generates music with a particular mood or 'affect expression' in response to player actions. The affective music generation system described in this paper is called MetaCompose. The goal of this research is to use MetaCompose for the synthesis of video-game music accompaniment.

Computer games have properties that make them particular interesting and challenging for this style of music generation: unlike traditional sequential media, such as novels or movies, events unfold in response to player input rather than a sequential and linear narrative. Therefore, a music composer for such an interactive environment needs to create music that is dynamic, while also holding the listeners' interest, and avoiding exactly repeating the piece, while maintaining a relationship to what has been heard before. This applies to a wide range of games although not all; e.g. rhythm games such as *Guitar Hero* make use of semi-static music around which the game-play is constructed [64,3].

This type of system is not limited to computer games; it can also be applied to any type of interactive or dynamic media: for example, adaptive (mood-based) music generation could be of interest for pervasive music [69,48,43,85], interactive artworks [31], or in the domain of ambient computing [20].

The purpose of MetaCompose is to produce music used as background in an interactive/dynamic experience. As such, it does not have to be as complicated and structured as, for example, a classical piece of music. MetaCompose has rather been designed to create small, loopable compositions with the focus on dynamically changing the music's affective expression. The affective moods are directed by an interactive application (e.g. a game) and the system has to be able to change between expressing moods fast enough to enable real-time interaction between the user, the game-play and the music. The system described here builds on experience with several earlier prototypes [81,82] which experimented with affective expression using simpler musical structures.

This paper extends a previously published conference paper [84] that describes the core parts of the music generator. The additional contributions in this extended paper include:

- Explanation of the affective expression layer (theory and implementation);
- Preliminary evaluation of the dimension of affect expression through dissonances;
- Description of the system in a greater detail, which we believe should be enough to reproduce this work.

The paper is structured as follows: Section 4 provides a concise high-level overview of the system's modules and how they relate. Section 5 describes the component-based framework for the generation of music abstractions, Section 6 specifies the details of the real-time music creation via affective expressive improvisation, and Section 7 describes the archive for storing music abstractions. In Section 8, the results of an earlier evaluation study are presented; these investigate the perceived quality of the music generated. Finally, in Section 9 the results of a second user study are presented, these investigate the expression of positive/negative mood through the introduction of dissonances.

2 Background

2.1 Music Generation and Games

Procedural generation of music is a field that has received much attention in the last decade [65]. The approaches are diverse and range from creating simple sound effects, to avoiding repetition when playing human-authored music, to creating more complex harmonic and melodic structures [25,26,19,18,46,2]. Wooller [94] divides approaches to procedural music generation into two categories, namely *transformational* and *generative* algorithms. Our music generator, MetaCompose, falls in the latter category since it creates music without having any predefined audio clips to modify or recombine.

Transformational algorithms act upon an already prepared structure, for example, by having music recorded in layers that can be added or removed at a specific time to change the feel of the music. Note that this is only an example and there are a many other transformational approaches [1,8], however a complete study of these is beyond the scope of this paper.

Generative algorithms instead create the musical structure themselves; this demands a higher degree of complexity in ensuring the music remains consistent, especially when connecting the music to game events. Such an approach requires more compute-power, as the music-content has to be created dynamically and on the fly. An example of this approach can be found in the game *Spore*: the music generator was created by Brian Eno with the *Pure Data* programming language [73], in the form of many small samples that recombine to create the soundtrack in real-time.

In this work, we adopt the latter approach, in particular focusing on generative procedural music composition in games for emotional expression. While the topics of affect [9], semiotics [30] and mood-tagging [58] are also interesting and significant, the focus of our system is *real-time generation of background music able to express moods.*

Many projects focus on expressing one (or more) affective states: an example is described by Robertson [75], where a music generator is developed to express fear. There are parallels between that work and our approach, for example musical data is represented via an abstraction (in their case via the CHARM representation [86, 93]), yet we claim our system has a higher affective expressiveness since it aims to express multiple moods in music. A more extensive example of a generative music system targeted at expressing particular emotions is described by Monteith [67] using Markov models, n-grams and statistical distributions sourced from a training corpus of music. Chan and Ventura's work [17], much like ours, focuses on expressing moods; yet their approach relies on changing the harmonization of a predefined melody, while ours generates the complete piece.

Evolutionary computation (EC) has a multitude of applications (see this recent review of the field [27]); while EC approaches have many features, we believe that the one that makes them particularly well-suited for creative tasks is that they are not focused on a single solution. Another aspect is that ECs are non-deterministic, if the problem is complex enough, ECs find different solutions based on the same initial conditions. There are many examples of evolutionary algorithmic (EA) approaches to generating music, two notable examples are the methods to evolve piano pieces by Loughran *et al.* [59] and Dahlstedt [20], although many others can be found in the *Evolutionary Computer Music* book [66]. Other examples of real-time music generation can be found in patents: two examples are a system that allows the user to play a solo over some generative music [74] and another that creates complete concert pieces in real-time [62]. An interesting parallel between the second system [62] and ours is the incorporation of a measure of "distance" between music clips in order to reduce repetition. Still, neither of the patented systems explicitly addresses affective expression.

As the final objective, our generator is designed to be employed to create computer game music. It is therefore important to mention the work by Livingstone [58], which defines a dynamic music environment where music tracks adjust in real-time to the emotions of the game character (or game state). While this work is interesting, it is limited by the usage of predefined music tracks for affective expression. Finally, another notable project in affective expressive music in games is *Mezzo* [12]: a system that composes neo-Romantic game soundtracks in real-time and creates music that adapts to emotional states of the character, mainly through the manipulation of *leitmotifs*.

2.2 Emotions and moods

Emotions have been extensively studied within psychology, although their nature (and what constitutes the basic set of emotions) varies widely. Numerous models of emotion have been developed since the seminal studies of the early 20th century [51, 79], arguably one of the most influential is the theory of basic or discrete emotions devised by Ekman [29, 28, 89]. The theory of basic emotions posits that all affective experiences derive from a core set of basic emotions which are distinct and independent. An alternate approach has been the development of dimensional models of affect, which posit that all emotions derive from a combination of two or more underlying psychological "dimensions [72, 80, 92]. Lazarus argues that "emotion is often associated and considered reciprocally influential with mood, temperament, personality, disposition, and motivation" [53]. Therefore the approach presented in this work aims to produce scores with an identifiable mood, and in so doing, induce an emotional response from the listener.

Affect is generally considered to be the experience of feeling or emotion. Brewin [11] states that affect is post-cognitive; namely emotions arise only after an amount of cognitive processing has been accomplished. With this assumption in mind, every affective reaction (e.g., pleasure, displeasure, liking, disliking) results from "a prior cognitive process that makes a variety of content discriminations and identifies features, examines them to find value, and weights them according to their contributions" [11]. Another view is that affect can be both pre- and post-cognitive, notably [55]; in this theory thoughts are created by an initial emotional response that in turn leads to an induced affect.

Moods are affective states. However, while an emotion generally has a specific object of focus, a mood tends to be more unfocused and diffuse [60]. Batson [6] posits that mood "involves tone and intensity and a structured set of beliefs about general expectations of a future experience of pleasure or pain, or of positive or negative affect in the future". Another important difference between emotions and moods is that moods, being diffuse and unfocused, may last longer [7].

In this paper, we focus on mood instead of emotion, for we expect that in games – where the player listens to the background music for a longer time – moods are more likely to be a determinant in player experience. In addition, they are easier for game designers to integrate, since they represent longer-duration sentiments, more suited to segments of game-play.

2.3 Moods in Music

Music is a powerful medium for affecting moods and this has been attested throughout history by poets, playwrights, composers, and researchers. Already Butler, in 1973, provided a bibliography of almost 900 publications in the 19th century that relate to the study of music psychology [14]. Unsurprisingly, the concept of mood expression and manipulation has been of great interest in the field of marketing; Bruner compiled an extensive, if now somehow dated, review of studies on music moods in connection to marketing [13]. It is important to note that there is disagreement about the kind of emotional responses evoked in the listener, and the alternative positions being argued that these are: i) "real" emotions, ii) a separate class of "aesthetic" emotions, or iii) moods [47]. We do not go in detail on this issue as the scope of this paper is limited to expressed – and not evoked – affective content.

The set of adjectives that can describe the mood of music and an emotional response to it is immense and there is no accepted standard vocabulary. For example, in the work of Katayose [44], the emotional adjective set includes *Gloomy, Serious, Pathetic* and *Urbane*. Other ontologies propose multiple mood clusters: eight in the case of Hevner [37] and ten in the case of Farnsworth [32]. Moreover, *All Music Guide*[1], a human-annotated music information database, uses an even larger descriptive approach with a total of 179 mood labels (not mutually exclusive).

[1] `http://www.allmusic.com`

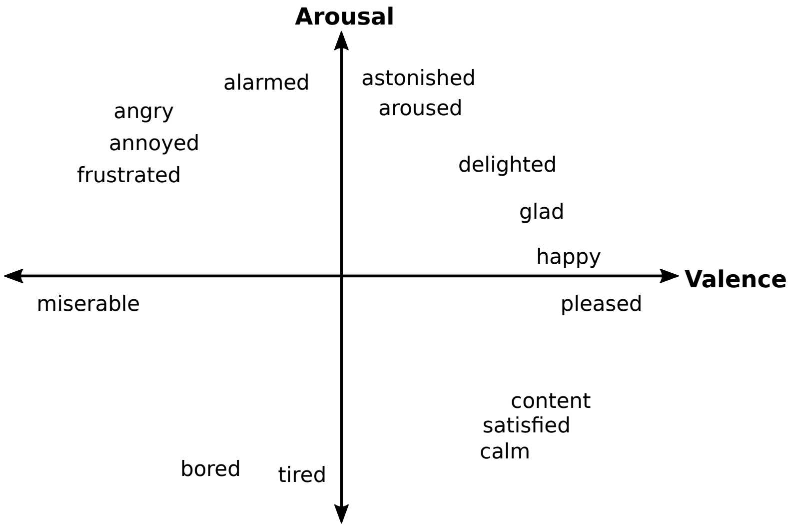

Fig. 1: The Valence-Arousal space, labeled by Russell's [77] direct circular projection of adjectives.

Russell [77] proposed a model of affect based on two bipolar dimensions: *pleasant-unpleasant* and *arousal-sleepy*, theorizing that each affect word can be mapped into a bi-dimensional space (Figure 1). Thayer [88] applied Russell's model to music using the dimensions of *valence* and *stress*; although the names of the dimensions are different from Russell's, their meaning is identical. Also, we find different terms among different authors [95,80] for the same moods. We use the terms *valence* and *arousal*, as they are the most commonly used in affective computing research. In this way, affect in music can be divided into quadrants based on the dimensions of valence and arousal: *Anxious/Frantic* (Low Valence, High Arousal), *Depression* (Low Valence, Low Arousal), *Contentment* (High Valence, Low Arousal) and *Exuberance* (High Valence, High Arousal). These quadrants have the advantage of being explicit and discriminate; also they are the basic music-induced emotions described in [49,56]. The model also views the axis as a continuous feature space, allowing for a theoretically unlimited combination of the two axis' expression. We adopted Thayer's approach as we believe its characteristics present more interesting computational applications (namely a continuous approach instead of a categorical one) and follow Russell's model of affect, still one of the most accepted in psychology.

Many attempts have been made to link emotions with specific aspects of music (tempo, mode, loudness, pitch, etc.). Hevner, in a classic series of studies, explored the effect of changes in mode, tempo, and pitch. Tempo and mode seem to be the strongest determinants of perceived emotion, yet the study had a flaw in the wording, making it unclear if the participants were annotating perceived or expressed affect [36,37,38]. Many more studies have been conducted since then, mainly exploring modes and tempo [50]. Far less explored are the associations with musical dimensions such as loudness, timbre [40] and pitch height [35] (for a review see [33]). A relatively unexplored area concerns the interplay of these dimensions, which are liable to be complex and somewhat idiosyncratic. Schellenberg conducted one such study, specifically on the interaction between rhythm and pitch [78]. Our mood expression theory (section 6.1) is based on the results of some of these previous studies. We also expect to find some interesting interplay between the various dimensions, which may shed some light in how features interact to convey affect.

In previous work, we built on these theories to evaluate affective expression in music through a crowd-sourced quantitative experiment: participants were asked to evaluate the affective expression perceived in the music composed through freeform answers [83]. Subsequently, words are stemmed (to group all the variations of similar words) and positioned in the bi-dimensional affective space through a best-localized criterion: the closer the words describing a part of the space are clustered, the more descriptive they are considered to be.

3 Methods

3.1 Multi-Objective Optimization

Multi-Objective Optimization (MOO) is defined as the process of simultaneously optimizing multiple objective functions. In most multi-objective optimization problems, there is no single solution that simultaneously optimizes every objective. In this case, the objective functions are said to be partially conflicting, and there exists, a number (possibly infinite) of Pareto optimal solutions. To understand what makes a solution better than another the concept of Pareto dominance is introduced: it is a binary relation between two solutions where one solution is Pareto dominant with respect to another solution if, for all objectives, it improves on the other solution. A solution is called "non-dominated", Pareto optimal, Pareto efficient or "non-inferior", if none of the objective functions can be improved in value without degrading one or more of the other objective values. Therefore, a practical approach to multi-objective optimization is to investigate a set of solutions (the best-known Pareto set) that represents the Pareto optimal set as much as possible [97]. Many Multi-Objective Optimization approaches using Genetic Algorithms (GAs) have been developed. The literature on the topic is vast; Coello lists more than 2,000 references on this topic on his website[2].

Our approach builds on the successful and popular NSGA-II algorithm [23]. The objective of NSGA-II is to improve the adaptive fit of a population of candidate solutions to a Pareto front, constrained by a set of objective functions. The population is sorted into a hierarchy of sub-populations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.

3.2 Feasible/Infeasible 2-Population Genetic Algorithm

Many search/optimization problems have not only one or several numerical objectives, but also a number of constraints – binary conditions that need to be

[2] http://www.cs.cinvestav.mx/˜constraint/papers/

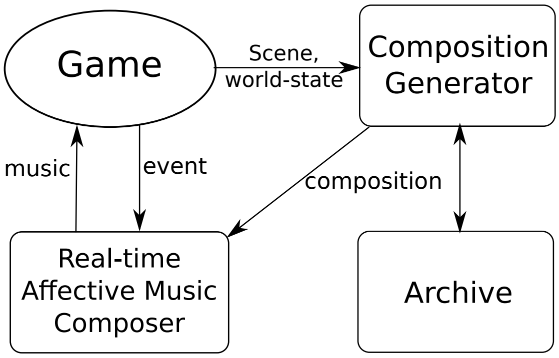

Fig. 2: METACOMPOSE's architecture.

satisfied for a solution to be valid. The approach we adopt for melody generation contains such strong rules; these are described in detail in Section 5.2.2. A number of constraint handling techniques have been developed to deal with such cases within evolutionary algorithms. The Feasible/Infeasible 2-Population method (FI-2POP) [45] is a constrained evolutionary algorithm that maintains two populations evolving in parallel, where feasible solutions are selected and bred to improve their objective function values, while infeasible solutions are selected and bred to reduce their constraint violations. In each generation, individuals are tested for constraint violations; if they present at least one violation they are moved to the 'Infeasible' population, otherwise they are moved to the 'Feasible' population. An interesting feature of this algorithm is that the infeasible population influences, and sometimes dominates, the genetic material of the optimal solution. Since the infeasible population is not evaluated by the objective function, it does not become fixed in a sub-optimal solution, but rather is free to explore boundary regions, where an optimum solution is most likely to be found.

3.3 Non-dominated Sorting Feasible-Infeasible 2 Populations

When dealing with constrained optimization problems, the approach is usually to introduce penalty functions to act for the constraints. Such an approach favors feasible solutions over the infeasible ones, potentially removing infeasible individuals that may lead to an optimal solution, and finding solutions that can be considered local optimum. There have been many examples of constrained multi-objective optimization algorithms [24, 16, 42, 41].

The internals of METACOMPOSE use a combination of FI-2POP and NSGA-II, dubbed Non-dominated Sorting Feasible-Infeasible 2 Populations (NSFI-2POP), which units the benefits of maintaining an infeasible population, free to explore the solution space without being dominated by the objective fitness function(s), and finding the Pareto optimal solution for multiple objectives. The algorithm takes the structure of FI-2POP, however the objective function of the feasible function is substituted with the NSGA-II algorithm. In Section 5.2 (below) an application of this approach to the evolution of melodies is described.

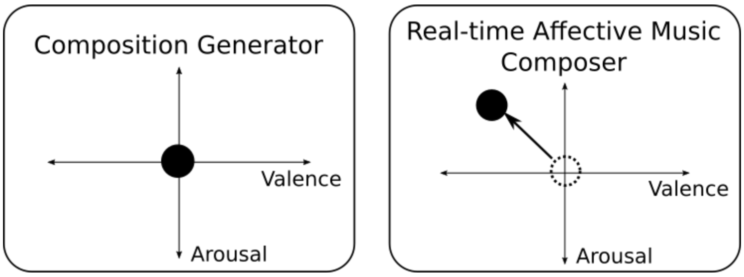

Fig. 3: The affective roles of the *composition generator* and of the *real-time affective music composer*: the first creates music abstractions that do not present any explicit affective expression, the latter renders compositions to reflect a specified mood.

4 MetaCompose

MetaCompose consists of three main components: (i) *composition generator* (Section 5), (ii) *real-time affective music composer* (Section 6) and (iii) an *archive* (Section 7) of previous compositions (Fig.2). The modular nature of MetaCompose allows components to be easily exchanged for others or augmented with further components.

The *composition generator* (i) creates the basic abstraction of music that will be used by the *real-time affective music composer* (ii) in order to create the final score according to a specific mood or affective state. In other words, as a metaphor, the *composition generator* (i) serves as a composer that only writes the basic outline of a piece, while the *real-time affective music composer* (ii) acts as an ensemble, free to interpret the piece in different ways. It is therefore important to note that the compositions generated by (i) do not include any explicit affective information, this is subsequently introduced by (ii) (see Figure 3). The purpose of this structure is to allow each composition to be expressed with potentially any affective content. The archive (iii) maintains a database of all the previous compositions connected to the respective levels/scenes of the game-state while also allowing a rank to be computed that measures the novelty of future compositions compared to those historically generated. MetaCompose is designed to be able to react to game events depending on the effect desired. Examples of responses to such events include: a simple change in the affective state, a variation of the current composition, or an entirely new composition.

MetaCompose is developed in Java[3], and makes use of the JMusic[4] and the Beads[5] [10] libraries.

[3] https://www.oracle.com/java/index.html
[4] http://explodingart.com/jmusic/
[5] http://www.beadsproject.net/

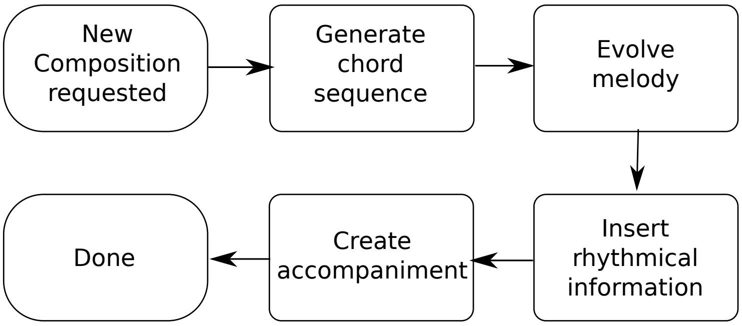

Fig. 4: Steps for generating a *composition*.

5 Composition Generation

Composition in this paper refers to an abstraction of a music piece composed by a *chord sequence*, a *melody* and an *accompaniment*. It is worth noting that the term *accompaniment* denotes an abstraction, not the complete score of a possible accompaniment, described in detail in Section 5.3 (below). The main reason for the deconstruction of compositions is to produce a general structure (an abstraction) that makes music recognizable and provides it with some referential identity. Generating abstractions, which themselves lack some information that one would include in a classically composed piece of music – e.g. tempo, dynamics, etc. – allows MetaCompose to modify the music played in real-time depending on the affective state the interactive media wishes to convey. The generation of compositions is a process with multiple steps: (i) creating a chord sequence, (ii) evolving a melody fitting this chord sequence, and (iii) producing an accompaniment for the melody/chord sequence combination (see Fig. 4).

5.1 Chord Sequence Generation

The method for generating a chord sequence works as follows: random walks are performed on a directed graph of common chord sequences (see Fig. 5) starting from a given chord. Referring to Figure 5, the graph does not use a specific key, but rather 'degrees': in music theory, a degree (or scale degree) is the name given to a particular note of a scale to specify its position relative to the 'tonic' (the main note of the scale). The tonic is considered to be the first degree of the scale, from which each octave is assumed to begin. The degrees in Fig. 5 are expressed in Roman numerals and, when talking about chords, the numeral in upper-case symbolizes a major chord, while lower-case (usually followed by an m) express a minor chord, which is sometimes omitted. Other possible variations on the chord are generally expressed with numbers and other symbols; these are not listed for the sake of brevity. Therefore, if we consider the D major scale, the $Dmajor$ chord would correspond to a I degree, while a $iiim$ degree would be a $F\sharp minor$. Various parameters of the generated sequence can be specified, such as sequence length, first element, last element, the chord to which the last element can resolve (e.g., if

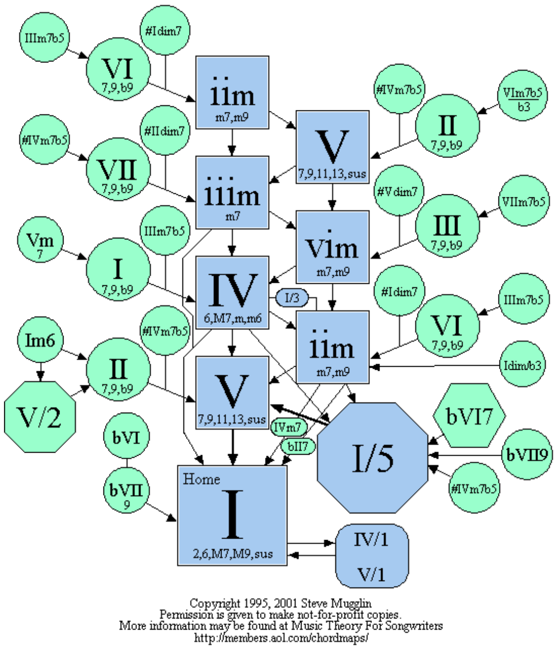

Fig. 5: Common chord progression map for major scales, created by Steve Mugglin [68].

we specify that we want the last chord to be able to resolve in the V degree, the last element might be a IV or a iim degree).

An interesting aspect of this graph is that it also shows common resolutions to chords outside of the current key, which provide a simple way of dealing with key changes. Each chord can be interpreted as a different degree depending on which key is considered, so if we want a key change we can simply: (i) find out which degree the last chord in the sequence will be in the new key and (ii) follow the graph to return to the new key. This should produce harmonious key changes that do not sound abrupt.

5.2 Melody Generation

Melodies are generated with an evolutionary algorithm approach. We define a number of features to include (objectives) and to avoid (constraints) in melodies, these are based on classical music composition guidelines and musical practice. These features are divided into constraints and objective functions. Accordingly, we use a Feasible/Infeasible two-population method (*FI-2POP* [45]) with multi-objective optimization [22] for the Feasible population. These features are in no way universally correct, but represent the current implementation of the system. It is possible to exchange them for others, and the system would still work, although the product quality would change for better or for worse. Given a chord sequence, a variable number of notes are generated for each chord, which will evolve without duration information. Once the sequence of notes is created, we generate the duration of the notes pseudo-randomly. We are easily convinced that there is potential to improve the system in this respect, and have experimented with different methods, yet we found no significant or obvious improvement compared to the method described. Therefore, instead of presenting a method which we do not believe creates a strong addition to the system, we present results with this simpler technique. Considering that these durations are only used as a starting point for the affective expression, we believe the impact on the music produced would be negligible.

5.2.1 Genome Representation

The evolutionary genome consists of a number of values (the number of notes to be generated) that can express the notes belonging to two octaves of a generic key (i.e. 0–13). Here, we do not introduce notes that do not belong to the key, effectively making the context in which the melodies are generated strictly diatonic. Variations will appear in later stages in the real-time affective music composer module when variations of the composition to express affective states or chord variations are introduced. The length of the genome depends on how many chords were generated in the previous step and can range from 1 to 8 notes per chord.

5.2.2 Constraints

We have three constraints. A melody should: (i) *not have leaps between notes bigger than a fifth*, (ii) *contain at least a minimum number of leaps of a second* (50% in the current implementation) and (iii) *each note pitch should differ from the preceding note pitch.*

$$\begin{aligned}&\mathit{Feasibility} = -\sum_{i=0}^{n-1}(\mathit{Second}(i,i+1) + \mathit{BigLeap}(i,i+1) + \mathit{Repeat}(i,i+1)),\\&\text{where } n \text{ is the genome length}\end{aligned} \tag{1}$$

The three functions comprising eqn. (1) are all Boolean that returning either 1 or 0 depending if the two notes at the specified indexes of the genome satisfy the constraint or not. As can be seen, this function returns a number that ranges from (potentially) $-3(n-1)$ to 0, where reaching the score 0 determines that the individual satisfies all the constraints and, consequently, can be moved from the infeasible population to the feasible population.

On the constraints in eqn. (1), leaps larger than a fifth do appear in music but they are avoided here, as experience suggests they can be hard on the ear of the listener [15]. Namely, if the listener is not properly prepared, leaps larger than a fifth can easily break the flow of the melody. We also specify a minimum number of intervals of a second (the smallest interval possible considering a diatonic context such as this, see the *Genome Representation* section) because if the melody has too many large leaps it feels more unstructured, not something that we would normally hear or expect a voice to sing. Finally, the constraint on note repetition is justified by the fact that repetitions will be introduced by the *real-time affective music composer*.

Fig. 6: An example of counter step-wise approach and departure from a leap (C-G).

5.2.3 Fitness Functions

Three objectives are used to compose the fitness functions: a melody should (i) *approach and follow big leaps (larger than a second) in a counter step-wise motion (explained below)* (eqn. 2), (ii) where the melody presents big leaps the *leap notes should belong to the underlying chord* (eqn. 3) and finally (iii) *the first note played on a chord should be part of the underlying chord* (eqn. 4).

First, we remind the reader that the definition of an interval in music theory is the **difference between two pitches**. In Western music, intervals are differences between notes belonging to the diatonic scale, e.g. considering a $Cmajor$ key, the interval between C and D is a *second*, the interval between C and E is a *third* etc.

$$\text{CStep} = \sum_{i=0}^{n-1} \left[\text{IsLeap}(i, i+1)(\text{PreCStep}(i, i+1) + \text{PostCStep}(i, i+1))\right] / leapsN$$

where $CStep$ stands for $CounterStep$, $PreCStep$ for $PreCounterStep$, and $PostCStep$ for $PostCounterStep$ (2)

$CStep$ (eqn. 2) measures *counter step-wise approach and follow to big leaps*. To clarify what *counter step-wise motion* means: if we examine a leap of a fifth from C to G, as in Fig. 6 – assuming we are in a $Cmajor$ key – this is a upward movement from a lower note to a higher note, a counter step-wise approach would mean that the C would be preceded by a higher note (creating a downward movement) with an interval of a second, therefore a D. Likewise, following the leap in a counter step-wise motion would mean that we need to create a downward movement of a second after the G, therefore we would need an F to follow.

The reason we introduce this objective is that it makes leaps much easier on the listener's ear, otherwise counter step-wise motions often sound too abrupt, by suddenly changing the range of notes the melody is playing. The `PreCStep` and `PostCStep` functions are Boolean functions that respectively check if the note preceding and following the leap approaches or departs with a contrary interval of a second.

The reason for having the leap notes – the two notes that form a leap larger than a second – as part of the underlying chord, is that such leaps are intrinsically more interesting than a step-wise motion, this means that the listener unconsciously considers them more meaningful and pays more attention to them [15].

When these leaps contain notes that have nothing to do with the underlying chord, even if they do not present real dissonances, they will be perceived as dissonant because they create unexpected intervals with the chord notes. Including leaps as part of the chord gives a better sense of coherence that the listener will consider as pleasant.

$$ChOnLeap = \sum_{i=0}^{n-1} \left[IsLeap(i, i+1)(BeToChord(i) + BeToChord(i+1)) \right] / leapsN,$$
where $ChOnLeap$ stands for Chord On Leap, and $BeToChord$
for Belongs To Chord (3)

The *ChOnLeap* function (eqn. 3) calculates how many *leap notes belong to the underlying chord* by checking if there is a leap between each two notes of the melody (*IsLeap*) and, if that is the case giving a positive score for each of the two notes that is part of the underlying chord ($BeToChord$ = belongs to chord).

$$FirstNtOnChord = \sum_{i=0}^{n} (IsFirstNote(i) \times BeToChord(i)) / chordsN$$
where $FirstNtOnChord$ stands for $FirstNoteOnChord$, and $BeToChord$ for $BelongsToChord$ (4)

The last objective (eqn. 4) emphasizes the importance of the first note following a chord change, by playing a note that is part of the chord we reinforce the change and make the chord change sound less discordant.

Note that these objectives, by the nature of multi-objective optimization, will generally not all be satisfied. This is acceptable, as satisfying all objectives might make the generated music sound too mechanical and predictable, while such "soft" rules are only enforced to a certain point, namely the Pareto frontier (contrary to the constraints of the infeasible population, which always need to be satisfied).

5.3 Accompaniment Generation

Accompaniment is included in the composition because, not only do chords and melody give identity to music, but also provide rhythm. Accompaniment is divided into two parts: a **basic rhythm** (a collection of note duration) and a **basic note progression** (an *arpeggio*). We can progress from the accompaniment representation to a score of the accompaniment by creating notes, with duration from the basic rhythm and pitches from the progressions (offset on the current underlying chord).

In previous work [84], we described an approach to generating rhythms consisting of a stochastic process involving combinations and modifications of some elements taken from a small archive of basic rhythms. MetaCompose was upgraded to make use of *Euclidean rhythms* [90], which have the property that their onset patterns are distributed as evenly as possible. Toussaint [90] also shows how such rhythms include a large family of rhythms used as rhythmic *ostinatos* in world music. Euclidean rhythms can be generated very efficiently and fulfill the objective of having a basic "pulse" for the instruments to follow.

Arpeggios are generated through a stochastic process involving combinations and modifications (inversions, mutations, etc.) of some elements taken from a small archive of basic rhythms. Specifically we have two basic arpeggios (see Fig. 7).
The algorithm performs the following steps:

Fig. 7: Basic arpeggios. These are represented as if they were played under a C major or C minor chord, but are transposed depending on what chord they appear underneath. Also the rhythmic notation of the arpeggio is dependent on the rhythmic structure.

1. choose a basic arpeggio;
2. shuffle the elements of the arpeggio;
3. increase the arpeggio to match the size of the basic rhythm (done by introducing at a random index of the arpeggio a new random pitch that already belongs to the arpeggio).

The rhythm presented in the final music will be modified by the real-time affective music composer for variety or for affect expression, while still maintaining a rhythmic and harmonic identity that will provide a characteristic signature for the composition.

5.3.1 Implementation details

This section serves to describe the specific operators and parameters used to obtain the results discussed later in the paper. The feasible population (i.e. that running NSGA-II) utilizes a **binary tournament** selection operator: two random individuals are chosen from the population and compared. The individual that dominates the pair is selected as a parent for a crossover operator, this operator is executed twice to obtain the two parents required by a crossover operator. In the event neither individual dominates the pair, a parent is chosen randomly among the pair. The infeasible population uses a **roulette-wheel** selection operator: the selection is a stochastic process where individuals have a probability of becoming parents for the next generation proportional to their fitness. In this way individuals with higher fitness are more likely to be selected while individuals with lower fitness have a lesser chance, however they may have genetic material that could prove useful to future generations and are therefore preserved. Both populations adopt a simple **single point crossover** operator and an **elitist strategy**, meaning that a specified number of the best individuals from the current population is allowed to carry on to the next one without being altered. The **mutation** operator gives each gene a probability $1/l$, where l is the genome length, to mutate. This ensures that on average only one gene will mutate but allows for more than one or no mutation to occur. The mutation itself transforms the note the gene represents to either the note directly above or directly below the mutated note in the scale.
The parameters used are:

– Population size: 500
– Generation number: 5000
– Elitist factor: 25%
– Mutation rate: 10%

6 Real-Time Affective Music Composer

The purpose of the real-time affective music composer is to go from the abstract compositions we have created to actual music. There are two main components of the process: a stochastic interpretation of the composition, that creates music that reflects the abstraction while presenting variations, and the modification of some musical and sonorous features to express affective meaning.

The system builds on a number of *instruments*, each with an algorithm to interpret some (or all) the data that comprises a composition; this interpretation is responsible for turning the abstraction into the score the system plays. These algorithms can be simple or complex, but should always contain some stochastic element, in order to avoid repetition. We created this instrument system to give some creative freedom to the developers that might use this music generation system; for example by making the instruments play accordingly to a specific music style or to create music that fits better with their accompanying media.

The instruments send the notes they generate in real-time to the standard MIDI device provided by the Java language. The system can be configured to be connected to an external MIDI device to provide for sound synthesis.

6.1 Mood expression theory

We now describe our model for mood expression in terms of music theory and how mood influences the production. We propose four musical features that influence perceived mood in music, these are: *intensity*, *timbre*, *rhythm*, and *dissonances*. These are mainly inspired by Liu *et al.* [57]. While Liu *et al.*'s research focused on mood classification via machine learning, we applied and expanded their model to generate music instead.

6.1.1 Intensity

Intensity (or Volume) is defined by how strong the volume of the music is. It is an arousal-dependent feature: high arousal corresponds to high intensity; low arousal to low intensity. Intuitively, high volume music results in increased stress. In a similar way, lower volume music, being less intense, is less arousing and has lower intensity.

6.1.2 Timbre

Timbre is defined as the combination of qualities of a sound that distinguishes it from other sounds of the same pitch and volume. For example, timbre is what makes the C4 chord sound different when played on a piano as opposed to a guitar. It is often associated with *"how pleasing a sound is to its listeners"* [4]. One of timbre's most recognizable features is what we could call "brightness", that is, how much of the audio signal is composed of bass frequencies. In previous literature audio features such as MFCC (Mel-Frequency Cepstral Coefficients [52] and spectral shape features [34] have been used to classify music on the basis of its timbral feature. We associated timbre with valence: the more positive the valence, the brighter the timbre.

6.1.3 Rhythm

Rhythm is divided into three features: strength, regularity and tempo [57].

- Rhythm strength: how prominent is the rhythmic section is (drums and bass). This feature is arousal dependent and our system acts by regulating the volumes of the instrument currently considered the "bass" to be proportionally higher or lower in the general mix.
- Regularity: how steady the rhythm is. This feature is valence dependent.
- Tempo: how fast the rhythm is. This feature is arousal dependent and influences the beats-per-minute (BPM) that the instruments follow.

In a high valence/high arousal piece of music, for instance, we observe that the rhythm is strong and steady. In a low valence/low arousal, on the other hand, the tempo is slow and the rhythm not as easily recognized.

6.1.4 Dissonance

Dissonance is the juxtaposition of two notes very close to each other: for example C and C♯. The distance between these two is just a semitone, which gives the listener a generally unpleasant sensation. A dissonant interval does not always sound bad. In fact most music contains dissonances, they can be used as cues expressing something amiss. The listener's ear can also be trained to accept dissonances through repetition, which explains why some music genres rely heavily on intervals that are avoided by other ones.In general, the larger the interval between the two dissonant notes, the 'easier' it is on the listener's ear: a C and a C♯ are always dissonant, but the dissonance is more evident if the notes are played from the same octave. C.P.E. Bach, in his *Essay on the True Art of Playing Keyboard Instruments* [5], remarks on the affective power of dissonances, although in a more general way: "... dissonances are played loudly and consonances softly, since the former rouse our emotions and the latter quiet them".

Meyer [63] observes that the affect-arousing role of dissonances is evident in the practice of composers as well as in the writings of theorists and critics, remarking how the affective response is not only dependent on the presence of dissonances *per se*, but also upon conventional association. This means that depending on the conventions of the musical style, dissonances might be more or less acceptable to the listener, and so can arouse different affective reactions. A study of listening preferences for infants, conducted by Trainor and Heinmiller [91], shows that even these young listeners, with no knowledge of musical scale, have an affective preference for consonance.

It is important to notice that, while dissonances do occur in music constrained to a certain key, we only consider the introduction of out-of-key notes to introduce a higher degree of dissonance in the music. We connect this feature to valence, hypothesizing that introducing more and more dissonances creates a more negative affect expression.

Out-of-key notes are introduced with the hypothesis that they create more negative valence. The more dissonances added, the more negative the valence. To maintain a feeling of purposeful composition, the possible notes that can appear correspond to the alteration of different scales, going further and further away from the Ionian mode (major mode) according to western music theory (this idea was inspired by Husain's study on mode's effects on mood [39]). It is important to note that MetaCompose does not perform proper mode changes; they are unprepared and the harmonic framework stays in a major key. Assuming a tonic of C, new notes that can appear in the piece are (from higher to lower valence):

- $B\flat$, from the Myxolydian mode
- $B\flat$ and $E\flat$, from the Dorian mode
- $B\flat$, $E\flat$ and $A\flat$, from the Aeolian mode
- $B\flat$, $E\flat$, $A\flat$ and $D\flat$, from the Phrygian mode

The reader might notice we are missing the Lydian and Locrian modes; we decided to exclude these scales for two different reasons. The Lydian mode is the same as the Ionian mode with its fourth degree raised, but it introduces a interval not as easily built upon to reach other modes (i.e. it requires the removal of the alteration to the fourth before adding new alterations). The Locrian mode is defined as a diminished mode as, although its third scale degree is minor, the fifth degree is diminished, instead of perfect. We have excluded it from our listening study mostly because we felt the music produced lost quality, as the diminished intervals make the piece sound less musically structured.

7 MetaCompose Archive

The archive's purpose is to store the previously generated compositions and associate these with game-related information, such as: levels, events or even entities. Another main function of the archive is the implementation of a distance measure that can give us information on how similar two compositions are. Such a measure helps manage diversity via similarity in previous and new tracks.

The distance measure we chose is the *Tonal Pitch Step Distance* [21], which measures distance of chord progressions on the basis of harmonic similarity. It uses a variant of Lerdahl's Tonal Pitch Space [54] to measure the distance of a chord to the tonic triad of its key. We have chosen this particular measure, even if it only considers the chord sequence part of our compositions, because we believe that a lot of the recognition of a song depends on the similarity of the chord sequence. Of course we realize that melodies are also very important, they are easy to recall and we generally connect them to a song, yet we believe that if you listen to the same melody, on two different chord sequences, it is not as immediately recognizable. On the other hand, the opposite is easier. In the future we might expand the distance measure to also include differences in the melody, but we decided it was not necessary for this first implementation of the system.

The main reason to have this mechanism is to allow the game-designer to direct MetaCompose to create music that the game-player would regard as consistent with the content presented to him/her previously. A case in which this might be useful is: if an important game-event occurs, which the game-designer wants to underline with a new composition and not just with a change in the affective meaning expression, he/she might want a new composition to share some similarity with the one previous heard and archived (e.g. the player might be in the same game-play level). As an example of the opposite, imagine the player moving from one area to another that is very different to anything previously seen. The game-designer may want to reflect this transition (at least partly) through music, so the game-designer directs MetaCompose to generate a track with a high distance score from any previously archived.

8 Evaluation of the Composition Generation

MetaCompose has been subject to an extensive quantitative study in order to validate our music generation approach. The main objective of the study is to investigate the contribution of each component of the framework to the perceived quality of the music created. To do this, MetaCompose components were systematically switched off and replaced them with random generation. From these random "broken" generators, and the complete MetaCompose system, we created various pair-wise samples to test against each other[6]. As the quality of music is a subjective matter, participants are asked to prefer one of two pieces of music presented to them, one generated by the complete algorithm and one from a "broken" generator with one component replaced with random generation. Quality is evaluated according to four criteria: *pleasantness*, *randomness*, *harmoniousness* and *interestingness*. These four criteria present a good overview of the preference expressed by the participant. Note that no definition of these terms is offered in the survey, and there is therefore no guarantee that participants interpret these criteria the same way (or for that matter differently).

Pleasantness intends to measure how pleasing to the ear the piece is, but this alone is not sufficient to describe the quality of the music produced. There are countless pieces of music that do not sound pleasant, but may nonetheless be considered by the listener as "good" music. In fact, in music, often uncommon (and even discordant) chord sequences or intervals are introduced to express different features in a score, such as affect as well as other narrative information. Also note that some alterations or passages can be specific of a music style. Moreover, discordant intervals are more acceptable to the ear the more often they are repeated they are (see dodecaphonic music [71] for example).

Interestingness is a criterion introduced to overcome the just described limitations of the *pleasantness* criterion: in this way we intend to test if one of our "broken" scores might introduce something that results in something considered interesting to the listener, even when the composition is not as pleasant or harmonic. Note that this is a very subjective measure, as most people have a different opinion about how interesting they perceive a score to be.

On the other hand, *harmoniousness* might be confused with pleasantness, but we hope that it will be seen as a somewhat more objective measure: less of a personal preference and more of a measure of the listener's ability to recognize the presence of dissonances and harmonic passages.

Finally, *randomness* intends to gathers a measure of how structured the music appears to the listener. It is not only a measure of dissonance (or voices being off-key), but also of how much the music seems to have a cohesive quality and coherent internal structure. Examples of coherent internal structure are: (i) voices working together well (ii) coherent rhythmic structure (iii) chord sequence presenting tension building and eventual resolution.

[6] This method is inspired by the "ablation studies" performed by Stanley [87].

An online survey was developed with HTML and PHP, using a MySQL database to hold the data collected. Participants were presented with pairs-wise music clips and asked to evaluate them using the four criteria described. Each of the four criteria has a multiple choices question structured as:

> **Which piece do you find more pleasing?** *"Clip A"/"Clip B"/"Neither"/ "Both Equally"*,

where the last word (e.g. "pleasing") is dependent on the criterion. We also include the more neutral answers "Neither" and "Both Equally" to avoid randomness in the data from participants who cannot decide which clip satisfies their evaluation according to the criterion better or worse. Other benefits of doing this are: avoiding the participant getting frustrated, and giving us some possibly information on interesting individual pairs, where the pieces are considered equally good/bad.

Note that, for the first five questions in the survey, the pair-wise clips always included one clip from the complete generator. After five trials, the clip pairs are picked at random between all the groups. In this way, we hoped to collect enough data to be able to make some observations between the four "broken" generators. The motivation behind this survey design choice is that our main question is evaluating the complete generator against all possible alternatives, so attention to the complete MetaCompose architectural has priority. This also has a practical justification in the fact that, with the number of groups we have (five), testing all possible combinations and gather enough data would be practically impossible. The survey has no pre-defined end: the user is able to continue answering until he/she wants, and can close the online survey at any time or navigate away from it without data loss. However, in the preamble to the survey, we encouraged participants to perform at least five comparisons.

8.1 Music clip generation

The five groups were examined (as dictated by the architecture of MetaCompose):

- **A. Complete generator:** the complete composition generator, MetaCompose, as described in Section 5;
- **B. Random chord sequence:** the chord sequence module is removed and replaced with a random selection of chords;
- **C. Random unconstrained melody:** the melody generation evolutionary algorithm is replaced with a random selection between all possible notes in the melody range (two octaves);
- **D. Random constrained melody:** the melody generation evolutionary algorithm is replaced with a random selection between all possible notes belonging to the key of the piece in the melody range (two octaves). We decided this was necessary as (by design) our melody evolution approach is restricted to a diatonic context;
- **E. Random accompaniment:** the accompaniment generation is replaced by a random accompaniment abstraction (we remind the reader that an accompaniment abstraction is defined by a basic rhythm and note sequence).

Table 1: Number of correct, incorrect and neutral answers to our criteria for the **complete generator** (A) against all the **"broken" generators** (B-E), combined. Note that in the case of the random criterion the listener is asked to select the clip that he/she feels the most random, so it is entirely expected that a low number of participants choose the **random clip** (E) against the **complete generator** (A).

Choice	Pleasing	Random	Harmonious	Interesting
MetaCompose (A)	654	197	671	482
Choose a "broken" generator (B-E)	240	633	199	327
A neutral answer	197	261	221	282
Total non-neutral answers	894	830	870	809
Binomial test p-value	7.44E-21	2.75E-77	2.05E-29	7.81E-02

For each of these 5 groups, 10 pieces of music are created. For the sake of this experiment the affect expression has been kept to a neutral state for all the groups and we used the same algorithms to improvise on the composition abstraction. There is therefore no exploration of the music generators' affect expression but rather an evaluation of the music quality from the complete architecture compared to the architectural alternatives. The clips for the various groups can be accessed at `http://msci.itu.dk/evaluationClips/`

8.2 Results and analysis

The data collected amounts to 1,291 answers for each of the four evaluation criteria from 298 participants. Of the survey trials generated, 1,248 contained a clip generated with MetaCompose (A). Table 1 shows how many responses were obtained for each criterion and how many neutral answers were collected.

For now we only consider definitive answers (where a participant chooses one of the music clips presented), we examine the impact of the neutral answers at the end of this section. Under this constraint, the data becomes Boolean: answers are either *"user chooses the clip from the complete generator (A)"* or *"user chose the clip from a broken generator (B-E)"*. To analyze this data we use a two-tailed binomial test, which is an exact test of the statistical significance of deviations from a theoretically expected random distribution of observations in two categories. The null hypothesis is that both categories are equally likely to occur and, as we have only two possible outcomes, that probability is 0.5.

8.3 Complete Generator against all other groups

Firstly, let us consider the combined results of all the "broken" groups (B-D) against MetaCompose (A): as can be seen from Tab. 1, we have statistically highly significant differences for the *pleasing*, *random* and *harmonious* categories, while we have a p-value of 0.078 for the *interesting* category. This means that we can refute the null hypothesis and infer a difference in distribution between choosing the music generated by the complete algorithm (A) and the "broken" ones (B-E).

Table 2: Answers and results of the binomial test for pairs comprised of the **full generator**, MetaCompose (A), and the one with **random chord sequences** (B).

MetaCompose (A) vs (B)	Pleasing	Random	Harmonious	Interesting
Successes	121	71	112	98
Failures	93	117	84	98
Totals	214	188	196	196
Binomial test p-value	3.23E-02	4.90E-04	2.68E-02	5.28E-01

We can affirm that MetaCompose (A) ranked better than all the others (B-E) for three of our four criteria, with the exception of *interestingness*, where there is no statistically significant difference. Interestingness is clearly a very subjective measure, and this may explain the result. Moreover, examining the ratio of neutral answers obtained for this criterion, it can be observed that it is almost 26%, a much higher neutral response that for the other criteria. This shows that in a higher number of cases participants could not say which composition they found more interesting. A possible explanation is that, as the affect expression (which also includes musical features such as tempo and intensity) is held in a neutral state, equal for all pieces, after hearing a number of clips listeners does not find much to surprise them. Also the duration of the generated pieces (ca. 30 seconds) might not allow sufficient time to determine interestingness.

8.4 Complete Generator against random chord sequence generation

If we only consider the pairs that included the MetaCompose (A) and the one with random chord sequences (B) (Tab. 2) we, again, obtain statistically significant differences in the distribution of the answers for the *pleasing*, *random* and *harmonious* criteria. In this case we have a very high p-value for *interestingness* (more than 0.5), in fact we have the same degree of preference for the MetaCompose (A) and the "broken" generator (B). We can explain this by considering that the disruptive element introduced by this modification of MetaCompose is mitigated by the fact that the rest of the system tries to create as pleasing music as it can, based on the chord sequence produced. So, for most of the time, the music will not have notes that sound out of key or that do not fit well with the chord sequence. Still, we observe how the listener is capable of identifying that, while the piece does not sound discordant or dissonant, it lacks the structure of tension-building and tension-releasing. This explains how MetaCompose (A) is preferred for all other criteria. It is interesting to note how the act itself of presenting the listener with uncommon chord sequences does result in an increase of the interestingness of the music.

8.5 Complete Generator against unconstrained melody generation

When we consider the unconstrained melody group we have statistically significant differences for all criteria, with some extremely strong significance (Tab. 3). These results are as we expected, as the melody plays random notes that conflict with both the chord sequence and the accompaniment.

Table 3: Answers and results of the binomial test for pairs comprised of the **full generator**, MetaCompose (A) and the one with **unconstrained random melody** (C).

MetaCompose (A) vs (C)	**Pleasing**	**Random**	**Harmonious**	**Interesting**
Successes	221	21	236	144
Failures	26	221	19	72
Totals	247	242	255	216
Binomial test p-value	5.15E-40	1.44E-43	4.11E-49	5.46E-07

Table 4: Answers and results of the binomial test for pairs comprised of the **full generator** MetaCompose (A), and the one with **constrained random melody** (D).

MetaCompose (A) vs (D)	**Pleasing**	**Random**	**Harmonious**	**Interesting**
Successes	125	81	120	108
Failures	100	109	85	94
Totals	225	190	205	202
Binomial test p-value	5.47E-02	2.49E-02	8.68E-03	1.80E-01

8.6 Complete Generator against constrained melody generation

The results given by the constrained random melody generation (D) are more interesting (Tab. 4). First, we notice no statistically significant values for the *pleasing* and *interesting* criteria. This is explained by the fact that the melody never goes off key, so it never presents off-key notes and never sounds abruptly "wrong" to the listener's ear. Yet, the *random* and *harmonious* criteria are statistically significant. Remembering how we described these criteria, we notice that the more objective criteria (*random* and *harmonious*) are those that demonstrate a difference in distribution. We believe this reinforces how, although compositions made in this group never achieve a bad result, the listener is still able to identify the lack of structure (randomness) and lack of consideration of the underlying chords of the melody (harmoniousness). An example of the first case would be a melody that jumps a lot between very different registers; this would make the melody sound more random than the melodies we evolve using (A) – MetaCompose – which follow more closely the guidelines of a singing voice. Harmoniousness can be influenced by the fact that, over a chord (expressed by the accompaniment), the melody can play notes that create intervals that 'confuse' the clarity of the chord to the listener's ear.

8.7 Complete Generator against random accompaniment generation

Finally, for the last group, the random accompaniment generation (E), gives us very clear statistically significant results on all criteria (Table 5). A lot of the harmony expression depends on the accompaniment generation, and when this is randomized it is no wonder that the piece sounds confusing and discordant. This is reflected in the trial data.

Table 5: Answers and results of the binomial test for pairs comprised of the **full generator**, MetaCompose (A), and the one with **random accompaniment** (E).

MetaCompose (A) vs (E)	Pleasing	Random	Harmonious	Interesting
Successes	188	25	203	132
Failures	21	186	12	63
Totals	209	211	215	195
Binomial test p-value	5.00E-35	6.58E-32	3.01E-46	4.35E-07

9 Evaluation of valence expression through dissonance

An evaluation conducted on an earlier prototype of a mood expressive music generator indicates that our mood expression theory better expresses arousal than valence (positive/negative feelings) [83]. Therefore, we are conducting a series of user studies to learn, in more depth, what the effect of the features we believe influence valence actually is. We present the results for an evaluation that focuses on the introduction of dissonant intervals by mean of altered tones (i.e. notes that are not included in the prevailing tonality, from now on also referred as out-of-key notes for brevity) and in particular how it affects valence in algorithmically generated music. The research questions this study addresses are:

1. Can negative valence be expressed via the introduction of altered tones in the generated music?
2. Can the quality of generated music be maintained when such notes are added?

To this end, we present and discuss the results of a participant-based evaluation study.

Traynor [91] shows how infants prefer consonant intervals to dissonant ones, yet there is a significant difference between instinctive preference and perceived negative valence. Therefore the connection between dissonance and positive/negative valence in MetaCompose is recorded and measured via an experimental platform where users listen to music generated by variations of an algorithm. Statistical analysis of user preferences is performed to characterize the differences between the generative clips in which examples of dissonant and consonant music are presented.

9.1 Experiment design

The main objective of this study is the evaluation of the valence expression from the introduction of altered tones in MetaCompose. The secondary objective is the maintenance of perceived music quality under these circumstances. An experiment was designed where pair-wise samples were tested against each other.

The survey asked participants to prefer one of two pieces of music presented and evaluate them according to three criteria: *most negative feeling expressed*, *most well-composed* and *most interesting*. The first criterion is the one to answer for our main research question: can negative valence by expressed by the introduction of altered tones? The other two ratings are more related to the secondary question: can we maintain the quality of the music while introducing altered tones?

We expected lower preference in the "well-composed" criterion for the *Out-of-key* group, as it can introduce intervals that might be unpleasant (although this might be subjective and sensitive to the cultural background of the individual). The "interestingness" criterion we expected to be mostly balanced, to show how the music with altered tones can still be interesting to listen to. Martinez and Yannakakis [61] suggest that ranking produces more consistent and reliable data when annotating affect information, therefore the choice of asking the participants to compare two pieces of music. Each criterion has a multiple choices question structured as:

> **Which piece seems to express more negative feelings?**
> *"Clip A"/"Clip B"/"Neither"/ "Both Equally"*,

where the wording is dependent on the three criteria. As in the study detailed in the previous section, we also include the more neutral answers "Neither" and "Both Equally" to avoid randomness in the data from participants who cannot decide which clip satisfies the evaluation criterion better or worse. The survey consists of ten questions, where the two clips presented are always one from the *consonant* and one from *dissonant* groups. The music pieces are chosen in a way that the participant will listen to all the clips produced for the experiment, but with random pairings between the two groups.

9.1.1 Music clip generation

Ten clips were created for each group (*consonant* and *dissonant*), for a total of 20 pieces. For the sake of this experiment the affect expression has been kept to a neutral state for all the mood-expressive features, apart from the *dissonances* feature. The same algorithms have been used to improvise on the composition abstraction. The clips for the various groups can be accessed at `http://msci.itu.dk/dissonanceClips/`.

Table 6: Participant's answers to our criteria. Also shown are the p-values, calculated using a two-tailed binomial test, and the Binomial Effect Size Display.

Choice	**Most negative**	**Most well-composed**	**Most interesting**
Altered group	329	118	191
Diatonic group	95	305	233
Neutral answer	112	113	112
Total not neutral answers	424	423	424
Binomial test p-value	1.38E-31	1.81E-20	2.32E-02
BESD	55.2%	-44.1%	-9.9%

9.2 Results and analysis

The data collected amounts to a total of 536 answers for each of the three evaluation criteria from 110 participants. Table 6 shows how many responses were obtained for each criterion and how many neutral answers were collected.

For now we only consider definitive answers (i.e. the participant chooses one of the music clips presented); we will look at the impact of the neutral answers at the end of this section. As for the previous study, under the definite choice constraint, the data becomes Boolean: the answers are either *"user chose the clip from the Out-of-key group"* or *"user chose the clip from the Diatonic group"*. To analyze this data we use a two-tailed binomial test, with as null hypothesis that both categories are equally likely to occur and, as we have only two possible outcomes, that probability is 0.5. The Binomial Effect Size Display (BESD) [76] is another way of looking on the effects of treatments by considering the increase of success through interventions. This is an interesting measure, as it elucidates how much of an effect is created, in our case, by the introduction of altered tones.

As it can be seen in Table 6, there is a strong statistical significance for the *most negative feeling*, *most well-composed* and *most interesting* categories. Thus the null hypothesis can be refuted and a difference in distribution can be inferred between choosing the music generated with (and without) the introduction of altered tones. This means that our system's introduction of out-of-key notes expresses more negative valence at the price of being perceived as less well-composed and less interesting. The BESD values reflect what can be inferred from the p-values, yet for the *most interesting* criterion the effect is much smaller (-9.9%), leading us to conclude that not as much interestingness is lost as the statistically significant p-value (2.32E-02) might suggest.

9.2.1 Outlier in dissonant group

An outlier was found by looking at the preference expressed for each of the *Out-of-key* music clips (see Figure 8). This specific clip has a very different distribution of answers than other *Out-of-key* clips with a much lower probability of being selected as more negative, a higher chance of being selected as more well-composed and a higher chance of being selected as more interesting. The piece is in $B\flat$ major, which would be composed of $B\flat$, C, D, $E\flat$, F, G and A. According to the system previously described, the piece should have its second, third, sixth and seventh degree altered by lowering them by a semitone, leading us to this scale: $B\flat$, $C\flat$, $D\flat$, $E\flat$, F, $G\flat$ and $A\flat$. Yet, we notice from the score that $C\flat$ and $G\flat$ never appear in the piece. This effectively removes all the strongest dissonances from the piece: those formed by notes distant by a semitone ($B\flat$-$C\flat$ and F-$G\flat$). This is a chance event, formed by a combination of both the way the *composition* was formed and the way the instruments have improvised over the abstraction.

By removing the data obtained by questions in which this clip appeared, we notice a slight increase in significance for all three criteria (in the order of 10^{-1}). This is expected, as it reinforces the distribution of the data we observed while considering all out-of-key samples.

9.2.2 Demographics

Our participant's population is composed by 89 males and 11 females. The average age is 29.5 (stdev 13.8). Participants were asked to rate their skill with a music instrument and their knowledge of music theory according to a five point Likert scale. The participants have reported a similar level of musical training (avg: 1.1 stdev: 0.97) and instrument skill (avg: 1.4 stdev: 1.2). The homogeneity of the

Fig. 8: Score for the outlier piece in the *out-of-key* group.

population may explain how, however we partition the population, we find no statistically significant difference in the answers given. We observed two participants that gave a high percentage of neutral answers (>75%). These participants self-reported to have close to no training and experience with music instruments. Yet the very low incidence of participants in this group makes it hard to make any assumptions, especially as many other participants reported a similar level of skill gave many definite choice answers.

If the population is divided by gender, we observe a higher preference in interestingness for dissonant pieces in males, yet the overall low number of female participants makes any conclusion here statistically unreliable.

10 Conclusion and future work

This paper described the MetaCompose component-based system for music generation based on creating an abstraction for musical structure that supports real-time affective improvisation. We exposed the method of creating the abstractions (*"compositions"*), which consists of the sequential generation of (i) chord sequences, (ii) melody and (iii) an accompaniment abstraction. The approach to evolve melodies using the non-dominated sorting with two feasible-infeasible populations genetic algorithm (NSFI-2POP) is also described in detail.

Two quantitative studies are also presented as an evaluation of the system: the first investigated the contribution of each component of the framework to the quality of the music created, while the second evaluated the valence expression of our dissonance introduction technique. In the first study we systematically switched off components of our generator and replaced them with random generation. From these random "broken" compositions and the from the complete algorithm architecture we created various pair-wise samples to test. In particular, we observed four broken groups: *random chord sequences*, *random melody constrained* (to the key of the piece), *random melody unconstrained* and *random accompaniment*. Analysis of the data supports the assertion that participants prefer the complete system in three of the four criteria: (*pleasantness*, *randomness* and *harmoniousness*) to the alternatives offered. The results for the *interestingness* criterion are however not definitive, but suggest that some parts of our generator have a higher impact on this criterion.

In the second study, analysis of the data supports the hypothesis that when altered tones are introduced in the generated music, participants evaluate the pieces as: i) expressing more negative valence; ii) being less well-composed and; iii) being less interesting. While we expected the samples containing out-of-key notes to be categorized as less well-composed, we hoped they would prove to be as interesting as consonant samples. We need to underline that, as can be seen by the Binomial Effect Size Display, the decrease in interestingness is only around 10%, so while the result is statistically significant we do not experience a great loss against the interestingness criterion. Moreover we realize that presenting both the "most well-composed" and "most interesting" questions might have been a mistake, as the participants might have been biased towards expressing an identical preference.

While the results in this paper show that we can reliably generate pleasant music capable of expressing specified moods, there are many further modules that could be added to MetaCompose, and many improvements that could be made. Future work will focus on evaluating the other valence-expressing music features, as in our previous study we found these to be perceived less consistently with our mood-expression theory [83]. After these studies we will evaluate the complete affect-expression capabilities of the system, following the methodology described by Scirea *et al.* for characterizing control parameters through crowd-sourcing [83]. Free-text description of the generated music will not only validate the expressive capabilities of the generator, but might also throw light of additional expressive capabilities beyond the two-dimensional model of emotions that we are currently using. Moreover a future study that should be conducted is comparing the music generated by MetaCompose to music created by a human. The intrinsic problem of this type of experiment is that MetaCompose presents a "compositional style" (i.e. small loopable compositions) and would very likely be very easy to distinguish from most human-composed music. This could create a polarization of the participants' answers, with a likely bias against the system. A solution would be to have the human compose in a similar style to the generator, yet this raises the ethical question of how much should we "constrain" the human. Finally we will integrate the system with multiple games and evaluate if any difference in play-experience can be observed. In summary, we have presented the detailed description of the MetaCompose system and we have shown: (i) how each part of our music generation method assists creating music that the listener finds more pleasant and structured; (ii) how our system for expressing valence through dissonance produces the expected perceived affective state and; (iii) presented a novel GA method for constrained multi-objective optimization.

References

1. Abrams, S., Oppenheim, D.V., Pazel, D., Wright, J., et al.: Higher-level composition control in music sketcher: Modifiers and smart harmony. In: Proceedings of the ICMC. Citeseer (1999)
2. Alpern, A.: Techniques for algorithmic composition of music **95**, 120 (1995). URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.9364&rep=rep1&type=pdf`
3. Arsenault, D.: Guitar hero:" not like playing guitar at all. Journal of the Canadian Game Studies Association **1**(2) (2008)
4. Aucouturier, J.J., Pachet, F., Sandler, M.: " the way it sounds": timbre models for analysis and retrieval of music signals. Multimedia, IEEE Transactions on **7**(6), 1028–1035 (2005)

5. Bach, C.P.E., Mitchell, W.J., John, W.: Essay on the true art of playing keyboard instruments. WW Norton (1949)
6. Batson, C.D., Shaw, L.L., Oleson, K.C.: Differentiating affect, mood, and emotion: Toward functionally based conceptual distinctions. (1992)
7. Beedie, C., Terry, P., Lane, A.: Distinctions between emotion and mood. Cognition & Emotion **19**(6), 847–878 (2005)
8. Biles, J.: Genjam: A genetic algorithm for generating jazz solos. In: Proceedings of the International Computer Music Conference, pp. 131–131. INTERNATIONAL COMPUTER MUSIC ACCOCIATION (1994)
9. Birchfield, D.: Generative model for the creation of musical emotion, meaning, and form. In: Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence, pp. 99–104 (2003)
10. Bown, O.: Experiments in modular design for the creative composition of live algorithms. Computer Music Journal **35**(3), 73–85 (2011)
11. Brewin, C.R.: Cognitive change processes in psychotherapy. Psychological review **96**(3), 379 (1989)
12. Brown, D.: Mezzo: An adaptive, real-time composition program for game soundtracks. In: Proceedings of the AIIDE 2012 Workshop on Musical Metacreation, pp. 68–72 (2012)
13. Bruner, G.C.: Music, mood, and marketing. the Journal of marketing pp. 94–104 (1990)
14. Butler, D.: An historical investigation and bibliography of nineteenth century music psychology literature. Ph.D. thesis, Ohio State University (1973)
15. Byron, T., Stevens, C.: Steps and leaps in human memory for melodies: The ef-fect of pitch interval magnitude in a melodic contour dis-crimination task. In: 9th International Conference on Music Perception and Cognition (ICMPC9), Bologna, Italy. Citeseer (2006)
16. Chafekar, D., Xuan, J., Rasheed, K.: Constrained multi-objective optimization using steady state genetic algorithms. In: Genetic and Evolutionary ComputationGECCO 2003, pp. 813–824. Springer (2003)
17. Chan, H., Ventura, D.A.: Automatic composition of themed mood pieces (2008)
18. Collins, K.: An Introduction to Procedural Music in Video Games. Contemporary Music Review **28**(1), 5–15 (2009). DOI 10.1080/07494460802663983. URL `http://dx.doi.org/10.1080/07494460802663983`
19. Cope, D.: Algorithmic Music Composition. In: G. Nierhaus (ed.) Patterns of Intuition, pp. 405–416. Springer Netherlands (2015). URL `http://link.springer.com/chapter/10.1007/978-94-017-9561-6_19`. DOI: 10.1007/978-94-017-9561-6_19
20. Dahlstedt, P.: Autonomous evolution of complete piano pieces and performances. In: Proceedings of Music AL Workshop. Citeseer (2007)
21. De Haas, B., Veltkamp, R.C., Wiering, F.: Tonal pitch step distance: a similarity measure for chord progressions. In: ISMIR, pp. 51–56 (2008)
22. Deb, K.: Multi-objective optimization using evolutionary algorithms, vol. 16. John Wiley & Sons (2001)
23. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on **6**(2), 182–197 (2002)
24. Deb, K., Pratap, A., Meyarivan, T.: Constrained test problems for multi-objective evolutionary optimization. In: Evolutionary Multi-Criterion Optimization, pp. 284–298. Springer (2001)
25. Doornbusch, P.: A brief survey of mapping in algorithmic composition. In: Proceedings of the International Computer Music Conference (2002). URL `http://www.academia.edu/download/33447946/A_Brief_Survey_of_Mapping_in_Algorithmic_Composition.pdf`
26. Edwards, M.: Algorithmic Composition: Computational Thinking in Music. Commun. ACM **54**(7), 58–67 (2011). DOI 10.1145/1965724.1965742. URL `http://doi.acm.org/10.1145/1965724.1965742`
27. Eiben, A.E., Smith, J.: From evolutionary computation to the evolution of things. Nature **521**(7553), 476–482 (2015)
28. Ekman, P.: Are there basic emotions? (1992)
29. Ekman, P.: An argument for basic emotions. Cognition & emotion **6**(3-4), 169–200 (1992)
30. Eladhari, M., Nieuwdorp, R., Fridenfalk, M.: The soundtrack of your mind: mind music-adaptive audio for game characters. In: Proceedings of Advances in Computer Entertainment Technology (2006)
31. Eno, B.: The ship. http://www.brian-eno.net/ (2016). Http://www.brian-eno.net/
32. Farnsworth, P.R.: The social psychology of music. (1958)

33. Gabrielsson, A., Juslin, P.N.: Emotional expression in music. Oxford University Press (2003)
34. Grey, J.M., Gordon, J.W.: Perceptual effects of spectral modifications on musical timbres. The Journal of the Acoustical Society of America **63**(5), 1493–1500 (1978)
35. Gundlach, R.H.: Factors determining the characterization of musical phrases. The American Journal of Psychology **47**(4), 624–643 (1935)
36. Hevner, K.: The affective character of the major and minor modes in music. The American Journal of Psychology **47**(1), 103–118 (1935)
37. Hevner, K.: Experimental studies of the elements of expression in music. The American Journal of Psychology **48**(2), 246–268 (1936)
38. Hevner, K.: The affective value of pitch and tempo in music. The American Journal of Psychology **49**(4), 621–630 (1937)
39. Husain, G., Thompson, W.F., Schellenberg, E.G.: Effects of musical tempo and mode on arousal, mood, and spatial abilities. Music Perception: An Interdisciplinary Journal **20**(2), 151–171 (2002)
40. Ilie, G., Thompson, W.F.: A comparison of acoustic cues in music and speech for three dimensions of affect. Music Perception: An Interdisciplinary Journal **23**(4), 319–330 (2006)
41. Isaacs, A., Ray, T., Smith, W.: Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. In: IEEE Congress on Evolutionary Computation, pp. 2780–2787. IEEE (2008)
42. Jimenez, F., Gómez-Skarmeta, A.F., Sánchez, G., Deb, K.: An evolutionary algorithm for constrained multi-objective optimization. In: Proceedings of the Congress on Evolutionary Computation, pp. 1133–1138. IEEE (2002)
43. Juslin, P.N., Liljeström, S., Västfjäll, D., Barradas, G., Silva, A.: An experience sampling study of emotional reactions to music: listener, music, and situation. Emotion **8**(5), 668 (2008)
44. Katayose, H., Imai, M., Inokuchi, S.: Sentiment extraction in music. In: Proceedings of the 9th International Conference on Pattern Recognition, pp. 1083–1087 (1988)
45. Kimbrough, S.O., Koehler, G.J., Lu, M., Wood, D.H.: On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. Eur. J. Operational Research **190**(2), 310–327 (2008)
46. Kirke, A., Miranda, E.R.: A Survey of Computer Systems for Expressive Music Performance. ACM Comput. Surv. **42**(1), 3:1–3:41 (2009). DOI 10.1145/1592451.1592454. URL `http://doi.acm.org/10.1145/1592451.1592454`
47. Konečni, V.J.: Does music induce emotion? a theoretical and methodological analysis. Psychology of Aesthetics, Creativity, and the Arts **2**(2), 115 (2008)
48. Krause, A.E., North, A.C., Hewitt, L.Y.: Music-listening in everyday life: Devices and choice. Psychology of music **43**(2), 155–170 (2015)
49. Kreutz, G., Ott, U., Teichmann, D., Osawa, P., Vaitl, D.: Using music to induce emotions: Influences of musical preference and absorption. Psychology of Music **36**(1), 101–126 (2008)
50. Krumhansl, C.L.: An exploratory study of musical emotions and psychophysiology. Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale **51**(4), 336 (1997)
51. Lange, C.G., James, W.: The emotions, vol. 1. Williams & Wilkins (1922)
52. Langlois, T., Marques, G.: A music classification method based on timbral features. In: ISMIR, pp. 81–86 (2009)
53. Lazarus, R.S.: Emotion and Adaptation. Oxford University Press (1991)
54. Lerdahl, F.: Tonal pitch space. Music Perception pp. 315–349 (1988)
55. Lerner, J.S., Keltner, D.: Beyond valence: Toward a model of emotion-specific influences on judgement and choice. Cognition & Emotion **14**(4), 473–493 (2000)
56. Lindström, E., Juslin, P.N., Bresin, R., Williamon, A.: "Expressivity comes from within your soul": A questionnaire study of music students' perspectives on expressivity. Research Studies in Music Education **20**(1), 23–47 (2003)
57. Liu, D., Lu, L., Zhang, H.J.: Automatic mood detection from acoustic music data. In: Proceedings of the International Symposium on Music Information Retrieval, pp. 81–7 (2003)
58. Livingstone, S.R., Brown, A.R.: Dynamic response: Real-time adaptation for music emotion. In: Proceedings of the 2nd Australasian Conference on Interactive Entertainment, pp. 105–111 (2005)

59. Loughran, R., McDermott, J., O'Neill, M.: Tonality driven piano compositions with grammatical evolution. In: IEEE Congress on Evolutionary Computation (CEC), pp. 2168–2175. IEEE (2015)
60. Martin, B.A.: The influence of gender on mood effects in advertising. Psychology & Marketing **20**(3), 249–273 (2003)
61. Martinez, H.P., Yannakakis, G.N., Hallam, J.: Dont classify ratings of affect; rank them! Affective Computing, IEEE Transactions on **5**(3), 314–326 (2014)
62. Meier, S.K., Briggs, J.L.: System for real-time music composition and synthesis (1996). US Patent 5,496,962
63. Meyer, L.B.: Emotion and meaning in music. University of Chicago Press (2008)
64. Miller, K.: Schizophonic performance: Guitar hero, rock band, and virtual virtuosity. Journal of the Society for American Music **3**(04), 395–429 (2009)
65. Miranda, E.R.: Readings in music and artificial intelligence, vol. 20. Routledge (2013)
66. Miranda, E.R., Biles, A.: Evolutionary computer music. Springer (2007)
67. Monteith, K., Martinez, T., Ventura, D.: Automatic generation of music for inducing emotive response. In: Proceedings of the International Conference on Computational Creativity, pp. 140–149. Citeseer (2010)
68. Mugglin, S.: Chord charts and maps. http://mugglinworks.com/chordmaps/chartmaps.htm. Accessed: 2015-09-14
69. North, A.C., Hargreaves, D.J., Hargreaves, J.J.: Uses of music in everyday life. Music Perception: An Interdisciplinary Journal **22**(1), 41–77 (2004)
70. Papadopoulos, G., Wiggins, G.: Ai methods for algorithmic composition: A survey, a critical view and future prospects. In: AISB Symposium on Musical Creativity, pp. 110–117. Edinburgh, UK (1999)
71. Perle, G.: Serial composition and atonality: an introduction to the music of Schoenberg, Berg, and Webern. Univ of California Press (1972)
72. Posner, J., Russell, J.A., Peterson, B.S.: The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and psychopathology **17**(03), 715–734 (2005)
73. Puckette, M., et al.: Pure data: another integrated computer music environment. Proceedings of the Second Intercollege Computer Music Concerts pp. 37–41 (1996)
74. Rigopulos, A.P., Egozy, E.B.: Real-time music creation system (1997). US Patent 5,627,335
75. Robertson, J., de Quincey, A., Stapleford, T., Wiggins, G.: Real-time music generation for a virtual environment. In: Proceedings of ECAI-98 Workshop on AI/Alife and Entertainment. Citeseer (1998)
76. Rosenthal, R., Rubin, D.B.: A simple, general purpose display of magnitude of experimental effect. Journal of educational psychology **74**(2), 166 (1982)
77. Russell, J.A.: A circumplex model of affect. Journal of Personality and Social Psychology **39**(6), 1161–1178 (1980)
78. Schellenberg, E.G., Krysciak, A.M., Campbell, R.J.: Perceiving emotion in melody: Interactive effects of pitch and rhythm. Music Perception: An Interdisciplinary Journal **18**(2), 155–171 (2000)
79. Scherer, K.R., Schorr, A., Johnstone, T.: Appraisal processes in emotion: Theory, methods, research. Oxford University Press (2001)
80. Schlosberg, H.: Three dimensions of emotion. Psychological review **61**(2), 81 (1954)
81. Scirea, M.: Mood dependent music generator. In: Proceedings of Advances in Computer Entertainment, pp. 626–629 (2013)
82. Scirea, M., Barros, G.A., Shaker, N., Togelius, J.: Smug: Scientific music generator. In: Proceedings of the Sixth International Conference on Computational Creativity June, p. 204 (2015)
83. Scirea, M., Nelson, M.J., Togelius, J.: Moody music generator: Characterising control parameters using crowdsourcing. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design, pp. 200–211. Springer (2015)
84. Scirea, M., Togelius, J., Eklund, P., Risi, S.: Metacompose: A compositional evolutionary music composer. In: International Conference on Evolutionary and Biologically Inspired Music and Art, pp. 202–217. Springer (2016)
85. Sloboda, J.A., O'Neill, S.A.: Emotions in everyday listening to music. (2001)
86. Smaill, A., Wiggins, G., Harris, M.: Hierarchical music representation for composition and analysis. Computers and the Humanities **27**(1), 7–17 (1993)
87. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2), 99–127 (2002)

88. Thayer, R.E.: The Biopsychology of Mood and Arousal. Oxford University Press (1989)
89. Tomkins, S.S.: Affect imagery consciousness: Volume I: The positive affects, vol. 1. Springer publishing company (1962)
90. Toussaint, G.T., et al.: The euclidean algorithm generates traditional musical rhythms. In: Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science, pp. 47–56 (2005)
91. Trainor, L.J., Heinmiller, B.M.: The development of evaluative responses to music:: Infants prefer to listen to consonance over dissonance. Infant Behavior and Development **21**(1), 77–88 (1998)
92. Watson, D., Tellegen, A.: Toward a consensual structure of mood. Psychological bulletin **98**(2), 219 (1985)
93. Wiggins, G., Harris, M., Smaill, A.: Representing music for analysis and composition. University of Edinburgh, Department of Artificial Intelligence (1990)
94. Wooller, R., Brown, A.R., Miranda, E., Diederich, J., Berry, R.: A framework for comparison of process in algorithmic music systems. In: Generative Arts Practice 2005 — A Creativity & Cognition Symposium (2005)
95. Wundt, W.: Outlines of psychology. Springer (1980)
96. Yannakakis, G.N., Togelius, J.: Experience-driven procedural content generation. IEEE Transactions on Affective Computing **2**(3), 147–161 (2011)
97. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation **8**(2), 173–195 (2000)