Velardo, Valerio and Vallati, Mauro

A Planning-based Approach for Music Composition

Original Citation

Velardo, Valerio and Vallati, Mauro (2015) A Planning-based Approach for Music Composition. In: Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research. CMMR 2015 . Springer, Plymouth, UK, pp. 778-790. ISBN 978-2-909669-24-3

Automated Planning and Music Composition: an Efficient Approach for Generating Musical Structures

Valerio Velardo[1] and Mauro Vallati[2]

[1] School of Music, Humanities and Media
University of Huddersfield, United Kingdom
valerio.velardo@hud.ac.uk
[2] School of Computing and Engineering
University of Huddersfield, United Kingdom
m.vallati@hud.ac.uk

Abstract. Automatic music composition is a fascinating field within computational creativity. While different Artificial Intelligence techniques have been used for tackling this task, Planning – an approach for solving complex combinatorial problems which can count on a large number of high-performance systems and an expressive language for describing problems – has never been exploited.
In this paper, we propose two different techniques that rely on automated planning for generating musical structures. The structures are then filled from the bottom with "raw" musical materials, and turned into melodies. Music experts evaluated the creative output of the system, acknowledging an overall human-enjoyable trait of the melodies produced, which showed a solid hierarchical structure and a strong musical directionality. The techniques proposed not only have high relevance for the musical domain, but also suggest unexplored ways of using planning for dealing with non-deterministic creative domains.

Keywords: Automatic music composition, Artificial Intelligence, Automated planning

1 Introduction

Automatic melodic composition is an important topic of computational creativity. In the last decades, numerous generative melodic systems that exploit Artificial Intelligence techniques have been developed [14]. Rader proposed a system based on a generative grammar whose rules are derived from fundamental music concepts [38]. Cruz designed a system which autonomously infers rules from a set of songs and employs them to generate new melodies [11]. Automatic melodic composition has also been interpreted as a constraint satisfaction problem [12, 36]. In this formalisation, the task of melodic generation corresponds to the solution of a problem with a number of musical constraints. To avoid the issue of encapsulating all the rules necessary to compose a melody directly within the system, numerous systems based on machine learning have been developed, which exploit either Markov Chains or Neural Networks [29, 41, 8, 9]. However, all these methods tend to produce melodies that are stuck in a single style. To avoid this pitfall, researchers have built several systems relying on optimisation techniques like genetic algorithms [23, 1, 22]. This approach simulates the biological evolutionary process based on reproduction, mutation, recombination and selection. Specifically, short melodic passages usually act as individuals of a population which are assessed by a fitness function. Cellular automata have also been employed to generate melodies, since they provide completely unpredictable musical results [30, 32]. The drawback of this approach though is that the aesthetic result is usually poor and the music produced can only be employed as raw material by human composers. Until now, to our knowledge a system that generates melodies based on Automated Planning has never been proposed, although planning – for its efficiency and its flexibility – could be a valuable support for automatic music composition.

In this paper, we try to fill this gap by introducing an AI system based on Automated Planning which focuses on the generation of musical structures. Specifically, we design two techniques, i.e., *flattening* and *tree*, that are able to produce a top-down musical structure for melodies, based respectively on a one-level and a multi-level hierarchical generative processes. AI planning, mainly because it is regarded as an approach for solving complex combinatorial problems, has seldom been considered for producing creative outputs, and never used for generating music. However, in the last years a few creative systems that exploit AI Planning have been proposed. In particular, research focused on developing effective methods for natural language generation [25], interactive storytelling and narrative planning [39, 37, 21]. Given the number of available high-performance planning engines and the expressivity of latest versions of the Planning Domain Description Language, PDDL [13, 19], there is no doubt that also automatic composition could benefit from creative systems based on planning. Specifically, we use planning as an efficient method for generating good quality musical structures. Indeed, a multiplicity of different structures can be obtained by simply leveraging on the innate randomness of some modules of planning algorithms. Produced musical structures are then filled by using a bottom-up technique [43], that provides raw musical materials. Evaluation of melodies has been carried out by music experts, that assessed the exploitability of the proposed planning-based approach.

Despite the strong focus on the music domain, we indirectly aim to demonstrate that the application of planning can be extended to all sorts of creative domains, which are usually based on non-deterministic processes. In this regard, planning could be an invaluable resource to efficiently help solve complex creative tasks, if used alongside other more traditional AI techniques in computational creativity.

In the remainder of this paper we initially provide the necessary background. Then the flattening and the tree techniques are described and their results are discussed. Finally, conclusions are given.

2 Background

In this section we describe Automated Planning, and the relevant musical information that are needed for understanding the proposed approach.

2.1 Automated Planning

Automated Planning studies the selection of actions in a dynamic system to reach a state of the system that satisfies a number of goals. Most of the existing approaches to Automated Planning assume that the system is deterministic, static, finite, and fully observable. Such a model is called *Classical Planning*, and mainly deals with finding a (partially or totally ordered) sequence of actions transforming environment from some initial states to a desired goal state [20]. A classical planning problem can be described as a tuple $\langle F, I, G, A \rangle$, where:

- F is a set of facts (or state variables) that represent the state of the world;
- I is a set of facts describing the initial state;
- G is a set of facts representing the goal;
- A is a set of actions, that represents the different possibilities of changing the current state, described by their preconditions and effects.

Solving a planning problem $\mathcal{P}$ consists of generating a plan, a sequence of actions $(a_1, a_2, \cdots, a_n)$ corresponding to a sequence of state transitions $(s_0, s_1, \cdots, s_n)$ such that: action a_i is applicable in state s_{i-1}, the state s_i is the result from executing a_i in s_{i-1} and s_n is a state where all goals are satisfied, $s_n \in G$.

The Classical Planning model can be extended, in order to handle a wider range of constraints and increase expressiveness. For instance, this is the case in Temporal Planning, where actions have a duration that should be considered, or Uncertainty Planning, that studies cases in which the environment is not fully observable and effects are non-deterministic. On this matter, the interested reader is referred to [20] and [17].

Likewise, each model has its own algorithms for effectively solving the corresponding planning problems. On the whole, one can say that state-of-the-art planners often rely on heuristic search to generate solutions [5]. Planning offers a large number of highly-efficient systems which are exploited in real-world applications. Moreover, the International Planning Competition – that is a major biennial event where planners are used for solving numerous problems – further fosters the design and development of new approaches.[3]

Although planning has been mainly exploited for addressing efficient problem-solving tasks, there are some academic studies that use the available planning systems for creative purposes. It is the case of narrative planning and interactive storytelling, which both exploit planners for automatically shaping stories [39, 37, 21, 2]. Recently, planning has been used also for generating grammatically correct sentences [25].

2.2 Music

Even though almost all humans are familiar with music, and have an intuitive understanding of what it is, there is no single agreed formal definition for it [34]. Definitions range from the idea of music as a subjective experience [3] and as organised sound [42], to the idea of music as a language [10], or as a social construct [33]. However, there are specific aspects of music, that although they are not sufficient to define it, would

[3] http://ipc.icaps-conference.org/

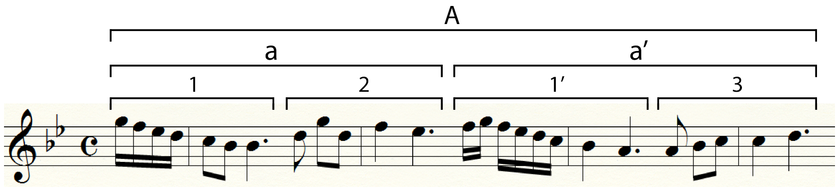

Fig. 1. Example of hierarchical structure in the first movement of the Sonata KV333 by Mozart. The top level sentence is divided into two double phrases, and four phrases.

be at least necessary for an operational definition. *Structure* is one of them. Specifically, music is based on hierarchical structures [28]. At the bottom level, a group of few notes considered together form meaningful musical clusters called *phrases*. Two or more phrases if combined form *double-phrases*. Two or more double-phrases together create musical *sentences*. The process happens recursively at several levels, until it reaches the level of the piece itself, intended as a whole. This phenomenon can be visualised as a tree (Figure 1). The top of the tree is the whole piece; intermediate levels represent sentences, double-phrases and phrases. Finally, the bottom level has single notes as constituents. However, the level of notes is not musically meaningful, since it does not carry enough musical information. As a consequence, the lowest musically meaningful level is the phrase level.

The hierarchical organisation of music is due to the way humans process stimuli they receive from the environment. In order to manage large amounts of information, humans group atomic elements together [35]. Group of groups are the next obvious step to manage even larger amounts of information. The grouping process continues at several levels, and it is carried out, because of memory constraints to which short-term memory is subject. Indeed, short-term memory, which is responsible for creating meaningful experiences of external stimuli on a real-time basis, can store 5 to 9 atomic elements at any level of abstraction only, for not more than 5 seconds [31]. For example, in the musical domain an atomic element can be a single tone, while in the visual domain an atomic element can be a single shape in an image. As Gestalt theory suggests, grouping happens both for sound and visual stimuli. Therefore, composers have always (un)consciously exploited grouping strategies, to easily transfer highly complex musical artifacts to listeners. Interestingly, the hierarchical organisation of music is an universal aspect of music, shared by all cultures all over the world [7]. This seems to imply that hierarchical organisation of music is embedded at a biological-cognitive level. It might be argued that some types of music – like aleatoric music composed by John Cage – do not have any hierarchical structure. This is an extreme case that nevertheless shows at least a low level structure provided by the first level grouping process (i.e., phrases), which people always tend to superimpose to any kind of music they listen to. Also, the very same action of choosing a specific strategy to generate music made by the composer – no matter how random it could be – inherently imposes a form of structure on the music itself, since all processes of generation involve a number of procedures which constrain the type of music that can be created. In other words, a generation strategy in-

```
(:action IS_EQUAL_TO
:parameters (?x ?y - phrases)
:precondition (and (is_before ?y ?x)
    (processed ?y)
    (clear ?x))
:effect (and (processed ?x)
        (not (clear ?x))
    (increase (total-cost) 2)))
```

Fig. 2. The operator used for identifying two musical phrases as equal, in the PDDL used by the flattening approach.

directly favours the selection of a specific set of musical patterns over the infinite others potentially available.

The hierarchical structure of music is also deeply connected to the way humans specifically process music. Listening to a piece of music can be regarded as a pattern matching task. While hearing music, listeners are able to identify musical phrases [24], to detect similarities between them, and to recognise new passages as well [26]. Consequently, from a generative point of view, composers have three valid strategies for creating music, i.e., repeating a passage, varying a passage, and adding a new passage [40]. This process can involve music elements belonging to any hierarchical level. Repeating a passage assures musical coherence, varying an element provides both unity and change, while introducing a new passage provides novelty. These methods if considered together are a perfect compositional strategy, which reflects basic cognitive elements we rely on, when processing external information. To create music, composers usually exploit two complementary approaches, i.e., *bottom-up composition* and *top-down composition* [43]. The former consists of filling the structures provided by the latter, with raw musical materials. In other words, the top-down strategy creates the hierarchical backbone of a piece, and the bottom-up approach provides the actual pitch, duration and timbre for each tone. Of course, the two strategies are linked together in a feedback chain, and can mutually influence themselves. The system we propose implements the top-down compositional process, generating structures which will be implemented as short melodies.

3 Musical Structure Generation as Planning

As already introduced, the composition process can be modelled as the combination of a top-down and a bottom-up approaches. In this work, we investigate how AI planning can be exploited in order to generate randomised, but still coherent, musical structures. For the aims of this work, the output of the musical structure generation process is, at each level of the tree shown in Figure 1, a sequence of elements and relations between them. An element can be either *new* or *similar* to a previous element. Similarity is represented by: (i) repeating or (iii) varying an element that comes before it in the sequence. While repetition is straightforward, we define variation as follows.

```
(IT_IS_NEW Ph2)
(IT_IS_NEW Ph1)
(IS_EQUAL_TO Ph8 Ph1)
(IS_EQUAL_TO Ph3 Ph2)
(IS_EQUAL_TO Ph9 Ph1)
(IT_IS_NEW Ph10)
(IT_MODIFIES Ph7 Ph3)
(IT_MODIFIES Ph4 Ph3)
(IT_IS_NEW Ph5)
(IT_MODIFIES Ph6 Ph4)
```

Fig. 3. An example solution plan resulting from a problem with 10 phrases, modelled following the flattening approach.

Definition 1. *An element e_1 is a variation of another element e if the following hold: (i) at least one constituent of e_1 is not a member of e, not even in a varied form; (ii) at least one constituent p_i of e_1 is either a repetition or a variation of a constituent p_j of e.*

It should be noted that both repetition and variation are commutative. If an element A is a repetition/variation of an element B, then B can be also considered as a repetition/variation of A.

For tackling the musical structure generation problem, we consider two alternatives: (i) flattening the tree structure on the lowest level, i.e., phrase level; and (ii) considering all structural levels during the generation process. In both cases, musical structure generation is encoded as a classical planning problem, by considering one or more specifically designed PDDL models. Thus, the proposed approaches are planner-independent and can potentially be used on a wide range of planning engines. It is worth noting that, in order to generate different structures, randomised planners should be used.

3.1 Flattening the structure

This approach allows to encode the musical structure generation process as a single planning problem, in which only musical phrases are considered. Clearly, such an approach does not fully exploit the potential of a structured composition. In particular, the lack of higher level structure limits the directionality of the melody, and could lead to musical pieces that sound more like random walks.

The key idea of the *flattening* approach is that every musical phrase is a specific problem object that has to be processed. A planning problem has a number of objects p_i of type phrases, that is defined by the user. The set of all phrases $P =< p_1, p_2, ...p_n >$ is an ordered sequence of musical phrases, that follows the order in which they will be played in the final melody. In PDDL, this is accomplished by using a (is_before ?x ?y) predicate, that specifies for every couple (p_i, p_j), with $i < j$, the relative order.

In the initial state, each phrase p_i is clear, and the goal is to have all phrases processed. A phrase can be processed by using three different operators, that correspond to generative actions that composers consider [40]: (i) introducing a new phrase, (ii)

`A, B, B, B′, C, B′′, B′, A, A, D`

Fig. 4. An example of a musical structure obtained by applying the proposed automated planning approach. Each letter corresponds to a different musical phrase. Apostrophes indicate varied versions of the corresponding original phrase.

repeating a phrase that appears before, and (iii) inserting a variation of another phrase. An example of the is_equal operator, which is used for repeating a phrase, is shown in Figure 2. In the PDDL model, a phrase can be equal to another phrase that appears before in the sequence, and that has been already processed. This is done in order to simulate as much as possible the human composition process: it is counter-intuitive to repeat something that is either undefined or not yet present in the sequence. Each operator has an associated cost, and the metric used in PDDL problems is to minimise the total cost. This provides some guide to planners in terms of operators that are preferred to be used, during the generative process.

The musical structure resulting is then extracted by analysing the solution plan provided by the planner. Figure 3 shows an example plan of a problem involving 10 musical phrases. By looking at the plan, we can derive that three different musical phrases are generated, due to three `IT_IS_NEW` actions; and are placed in positions 1, 2 and 10 of the sequence. Such phrases are either repeated (`IS_EQUAL_TO` actions) or varied (`IT_MODIFIES`) in different positions of the sequence. The corresponding structure is shown in Figure 4. Each letter corresponds to a different musical phrase. Apostrophes indicate varied versions of the corresponding original phrase, and can be recursively applied. The structure does not indicate *how* two similar phrases differ; this is demanded of the bottom-up approach, that will then fill the "empty" structure with raw musical material.

3.2 Modelling the whole tree structure

In this approach, we tackle the problem of exploiting automated planning for generating three different levels of musical structures: namely, sentences, double phrases and phrases (see Figure 1). Intuitively, decisions taken for higher levels of the hierarchy affect and constraint significantly the shape and size of lower ones. For instance, a repeated sentence has the same double phrases of the initial model, but this is not true for a varied sentence. Encoding in a single planning problem the process of generating three levels poses a number of non-trivial issues. A relevant issue is the fact that a number of objects across the musical tree dynamically change, following the shaping process of a step. Moreover, one level has to be fully processed before considering the lower one. Also, a PDDL model that encodes such a complex domain at once would be –from a knowledge engineering perspective– extremely hard to validate, verify and maintain. Finally, performance of domain-independent planners on such problems can be hard to predict.

Given the identified issues, we opt for a modular approach. Every level of the musical tree structure is encoded as a different PDDL problem. Plans generated for solving problems that refer to higher levels of the structure affect the number of objects and

```
(:action SIMILAR_EQUAL
:parameters
(?dpx ?dpy ?dpz - doubleph ?sx ?sy - sentence)
:precondition (and
(clear_similar ?dpx ?sx ?sy)
(clear_similar ?dpy ?sx ?sy)
(different ?dpx ?dpy)
(is_before ?dpz ?dpx)
        (member_of ?dpz ?sx)
        (member_of ?dpx ?sy)
        (member_of ?dpy ?sy)
(processed ?dpz))
:effect (and
(processed ?dpx)
(processed ?dpy)
(increase (total-cost) 2)))
```

Fig. 5. The operator used for processing two double phrases from a sentence *?sy* which is similar to *?sx* .

the initial states of subsequent problems. Each level is encoded using PDDL models that are similar to the one described for the "flattening" approach. The main differences are: (i) the dependency between problems; and (ii) special operators for dealing with objects that are constituents of repeated or varied higher-level elements. Let us describe them by considering an example. For instance, in the current problem we want to generate the musical structure at the double phrases level. Thus, sentences have already been processed. For the purposes of this example, let us consider sentences shaped as follows: `A,B,A',B`. In order to generate the corresponding double phrases problem, constituents of each sentence have to be identified. In particular, sentence 2 and 4 must have the same number of double phrases –as well as exactly the same double phrases–, while sentences 1 and 3 can have a different number of double phrases ($2 \leq N \leq X$, where X is usually 3), regardless of the fact that sentence 3 is a variation of the first one. This directly derives from Definition 1.The number of double phrases for each sentence is randomly decided, following the aforementioned constraints. In the initial state of the problem, double phrases are associated with the corresponding sentence using the member_of predicate. Double phrases that are part of repeated sentences are forced to be equal by a specific predicate, that pairs them and allows only an equal operator to be used for processing them. Finally, in the case of a varied sentence, Definition 1 is followed, by forcing either `SIMILAR_MODIFY` or `SIMILAR_EQUAL` operators (depicted in Figure 5) to be used on double phrases of such varied sentence. These operators processes two double phrases at a time; the latter by repeating a double phrase from `?sx` into `?sy`, while the former includes a variation of a double phrase. The `clear_similar` predicate indicates that a double phrase is a constituent of a sentence `?sx` that is similar to `?sy`.

4 Experimental Evaluation

The approaches depicted in the previous section have been used for composing melodies in order to evaluate their musical outputs. In this section we firstly describe the experimental setup, and we then discuss the results of the empirical evaluation.

4.1 Experimental Setup

Musical Structures have been generated by using the well-known domain-independent planner LPG [18] on problems from the two approaches. LPG is a versatile system that can be used for plan generation, plan repair and incremental planning [16]. The planner is based on a stochastic local search procedure that explores a space of partial plans represented through *linear action graphs*, which are variants of the very well-known planning graph [4]. LPG has been chosen due to its high degree of randomisation – it is based on local search, and the seed can be set by the user – and to its ability to provide solutions of increasing quality, according to the used metrics. We manually set the cost of each action of PDDL domain models, in order to obtain proper structures. In particular, we assign a very low cost to the variation operator, with regards to the repetition operator. It should be noted that significantly different structures can be obtained by modifying only the cost associated to operators. The generated structures were then filled using the bottom-up technique proposed in [43]. This method relies on a Markovian process, trained on a corpus of stylistically similar melodies, that allows to generate musical phrases of 5 to 10 notes, and to use and modify them for filling the structure provided by our planning-based top-down approach. In our experiments the training has been done using the German folk songs of the well-known Essen database.[4]

We generated the structure of 10 short melodies – 70-90 seconds long – for each of the two planning-based strategies (i.e., flattening and tree).[5] The process took about a CPU-time second for each melody. In order to evaluate the quality of structures, we asked three music experts to assess the 20 short melodies, filling a questionnaire with quantitative and qualitative questions. Experts have a Phd in music-related topics and more than ten years of experience in music composition. We asked experts to rate the overall quality of the melodies with a scale from 1 to 10, where 1 indicates minimum aesthetic value and 10 stands for maximum aesthetic value. Raters provided also qualitative comments assessing the structure of the melodies they heard. To put evaluation in context, the experts evaluated 20 melodies generated with the two proposed planning-based systems and 10 generated following the approach in [43], which exploited a random-walk top-down algorithm for the structure. The reliability of agreement between experts has been evaluated through Fleiss' kappa test [15]. The scores provided by the three experts are overall quite consistent, since inter-rater reliability is $k = 0.43$ which indicates moderate agreement among raters [27].

[4] http://www.esac-data.org/

[5] Example of generated melodies can be found at: http://helios.hud.ac.uk/scommv/storage/example.zip

4.2 Results

The differences between the scores of the melodies generated with the three systems allow us to gain some interesting insights into the effectiveness of the planning-based approaches for the generation of musical structures. Indeed, all three systems evaluated rely on the same algorithm to fill the structure, therefore the only difference between them lies in the strategy adopted to produce the structure. The means of the scores are quite similar in the cases of melodies generated with the tree approach and with the random-walk strategy: $s_t = 6.5$, $s_{rw} = 6, 6$. On the other hand, the mean of the scores of the melodies created with the flattening approach is remarkably lower, $s_f = 2.3$. We used a significance level of $\alpha = 0.05$ for statistical analysis. The result of a Student's t-test conducted to compare the mean scores of the melodies generated with the flattening approach and those produced with the random-walk strategy confirms that there is a significant statistical difference between the two sets of scores, $t(9) = 3.536, p < 0.005$, two-tailed. A second Student's t-test shows that there is no significant statistical difference between the scores obtained by the structure generated with the random-walk methodology and those composed with the tree approach ($t(9) = 0.708$, $p > 0.05$, two-tailed). In other words, the tree approach is as effective as the original strategy adopted in [43] to generate musical structures. The flattening strategy instead produces completely unsatisfactory melodies. The low scores of the melodies composed with the latter system should be attributed only to the poor structure of the melody. The difference in scores between different strategies supports the initial statement that structure plays a major role for developing good melodies and that it can be considered in isolation, separated from other musical elements such as harmony, style and rhythm.

Music experts agree that melodies generated with the tree top-down approach show a strong organisation. Melodies are built upon musical elements that are nicely repeated and modified. Also, the balance between materials already heard and new ones usually works quite well. The result is a group of melodies that show an overall sense of unity, with pleasant elements of novelty. A strong hierarchical organisation of musical elements contributes to create a human-like feeling with clear directionality. On the other hand, melodies generated with the flattening strategy show a weak structure. Phrases appear to be randomly repeated and modified. According to one music expert, the organisation of these passages is not different from a complete random process. The structure of melodies composed with the tree approach and that of the Velardo and Vallati ([43]) study result overall quite similar. However, two out of three experts noticed that melodies generated with planning have often a simpler structure, than that of melodies composed exploiting the random-walk top-down approach. Specifically, an expert claimed that melodies generated with planning show a structure "similar to those of some folk songs".

5 Conclusion

Automatic generation of music is a challenging subfield of computational creativity. Many approaches of Artificial Intelligence have been used, to develop systems able to create music enjoyable by humans. However, due to its intrinsic deterministic way of managing problems, planning has never been adopted for music creative systems.

In this paper, we proposed the first method we are aware of which creates musical structures by exploiting automated planning techniques. The approach is modular and planner-independent, thus can exploit any planning system to generate music; moreover, it has been designed in order to be easy to extend, by incorporating more sophisticated operators or musical constraints. Specifically, we developed two methods, i.e., flattening and tree, that are able to produce a top-down musical structure for melodies, based respectively on one-level and multi-level hierarchical generative processes. These approaches are hugely scalable, since they can produce structures for hours of music in a few seconds. Our experimental analysis, which involved three musical experts and several short melodies generated by the proposed approaches, indicates that the tree approach is quite effective for generating human-like melodies, while the flattened method is not. In particular, experts highlighted that the flattened method leads to melodies with weak structure, with phrases that appear to be randomly positioned.

This work does not only impact the music domain, but can also open new avenues of research in planning, since it suggests innovative unexplored ways of using planning techniques to deal with non-deterministic creative outputs. Future work will introduce an evolutionary trait in the system, which will allow an ongoing change in the way the structures are generated, by considering preferences during the planning processes or modifying cost associated to each operator. We are also interested in alternative encodings of the musical structure generation problem; for instance by considering continuous planning or planning with preferences. Remarkably, automated planning can be fruitfully employed in Music Information Retrieval tasks. For instance, the large number of available highly efficient planning systems can be exploited in the automatic playlist generation field [6]. Finally, we want to develop a general framework for iteratively dealing with complex multi-level structures from different artistic domains, like fractals, visual art, or complex story generation.

References

1. Alfonseca, M., Cebrián, M., Ortega, A.: Evolving computer-generated music by means of the normalized compression distance. WSEAS Transactions on Information Science and Applications 2, 1367–1372 (2005)
2. Barros, L.M., Musse, S.R.: Planning algorithms for interactive storytelling. Comput. Entertain. 5(1) (Jan 2007)
3. Berio, L., Dalmonte, R., Varga, B.A., Osmond-Smith, D.: Two interviews. M. Boyars (1985)
4. Blum, A.L., Furst, M.L.: Fast planning through planning graph analysis. Artificial Intelligence 90, 281 – 300 (1997)
5. Bonet, B., Geffner, H.: Planning as heuristic search. Artificial Intelligence 129, 5–33 (2001)
6. Bonnin, G., Jannach, D.: Automated generation of music playlists: Survey and experiments. ACM Computing Surveys (CSUR) 47(2), 26 (2014)
7. Brown, S., Jordania, J.: Universals in the worlds musics. Psychology of Music 41(2), 229–248 (2013)
8. Browne, T.M., Fox, C.: Global expectation-violation as fitness function in evolutionary composition. In: Applications of Evolutionary Computing, pp. 538–546 (2009)
9. Coca, A.E., Romero, R.A., Zhao, L.: Generation of composed musical structures through recurrent neural networks based on chaotic inspiration. In: Proceedings of the International Joint Conference on Neural Networks. pp. 3220–3226 (2011)

10. Cooke, D.: The language of music (1959)
11. Cruz-Alcázar, P.P., Vidal-Ruiz, E.: Learning regular grammars to model musical style: Comparing different coding schemes. In: Grammatical Inference, pp. 211–222 (1998)
12. Davismoon, S., Eccles, J.: Combining musical constraints with markov transition probabilities to improve the generation of creative musical structures. In: Applications of Evolutionary Computation, pp. 361–370 (2010)
13. Edelkamp, S., Hoffmann, J.: Pddl 2.2: The language for the classical part of the 4th international planning competition, albert ludwigs universität institüt fur informatik, freiburg. Tech. rep., Germany, Technical Report (2004)
14. Fernández, J.D., Vico, F.J.: AI methods in algorithmic composition: A comprehensive survey. J. Artif. Intell. Res. (JAIR) 48, 513–582 (2013)
15. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychological bulletin 76(5), 378 (1971)
16. Fox, M., Gerevini, A., Long, D., Serina, I.: Plan stability: Replanning versus plan repair. In: Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling (ICAPS 2006). pp. 212 – 221 (2006)
17. Geffner, H., Bonet, B.: Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2013)
18. Gerevini, A., Saetti, A., Serina, I.: Planning through stochastic local search and temporal action graphs in LPG. J. Artif. Intell. Res. (JAIR) 20, 239–290 (2003)
19. Gerevini, A.E., Haslum, P., Long, D., Saetti, A., Dimopoulos, Y.: Deterministic planning in the fifth international planning competition: Pddl3 and experimental evaluation of the planners. Artificial Intelligence 173(5), 619–668 (2009)
20. Ghallab, M., Nau, D., Traverso, P.: Automated planning: theory & practice. Elsevier (2004)
21. Haslum, P.: Narrative planning: Compilations to classical planning. J. Artif. Intell. Res. (JAIR) 44, 383–395 (2012)
22. Jensen, J.H.: Evolutionary music composition: A quantitative approach. Ph.D. thesis (2011)
23. Johnson, M., Tauritz, D.R., Wilkerson, R.: Evolutionary computation applied to melody generation. In: Proc. of the Artificial Neural Networks in Engineering (ANNIE) (2004)
24. Knösche, T.R., Neuhaus, C., Haueisen, J., Alter, K., Maess, B., Witte, O.W., Friederici, A.D.: Perception of phrase structure in music. Human Brain Mapping 24(4), 259–273 (2005)
25. Koller, A., Hoffmann, J.: Waking up a sleeping rabbit: On natural-language sentence generation with FF. In: Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS. pp. 238–241 (2010)
26. Krumhansl, C.L.: A perceptual analysis of mozart's piano sonata k. 282: Segmentation, tension, and musical ideas. Music Perception pp. 401–432 (1996)
27. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics pp. 159–174 (1977)
28. Lerdahl, F., Jackendoff, R.: A generative theory of tonal music. MIT press (1985)
29. Lo, M.Y., Lucas, S.M.: Evolving musical sequences with n-gram based trainable fitness functions. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 601–608 (2006)
30. Millen, D.: Cellular automata music. In: Proceedings of the 1990 International Computer Music Conference. pp. 314–316 (1990)
31. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological review 63(2), 81 (1956)
32. Miranda, E.R.: Cellular automata music: An interdisciplinary project. Journal of New Music Research 22, 3–21 (1993)
33. Nattiez, J.J.: Music and discourse: Toward a semiology of music. Princeton University Press (1990)

34. Nettl, B.: An ethnomusicologist contemplates universals in musical sound and musical culture. The origins of music pp. 463–472 (2000)
35. Paas, F., Renkl, A., Sweller, J.: Cognitive load theory and instructional design: Recent developments. Educational psychologist 38(1), 1–4 (2003)
36. Pachet, F., Roy, P., Barbieri, G., Paris, S.C.: Finite-length markov processes with constraints. transition 6(1/3) (2001)
37. Porteous, J., Teutenberg, J., Charles, F., Cavazza, M.: Interactive storytelling with temporal planning. In: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3. pp. 1323–1324. International Foundation for Autonomous Agents and Multiagent Systems (2011)
38. Rader, G.M.: A method for composing simple traditional music by computer. Communications of the ACM 17(11), 631–638 (1974)
39. Riedl, M.O., Young, R.M.: Narrative planning: Balancing plot and character. J. Artif. Intell. Res. (JAIR) 39, 217–268 (2010)
40. Strang, G., Schönberg, A.: Fundamentals of musical composition. Faber & Faber (1967)
41. Thornton, C.: Hierarchical markov modeling for generative music (2009)
42. Varèse, E., Wen-Chung, C.: The liberation of sound. Perspectives of new music pp. 11–19 (1966)
43. Velardo, V., Vallati, M.: Automatic melody composition and evolution: a cognitive-based approach. In: Proceedings of the 9th Conference on Interdisciplinary Musicology CIM (2014)