

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

Submission 73

Abstract

Automatic music composition is a fascinating field within computational creativity. While different artificial intelligence techniques have been used for tackling this task, Planning –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.

Introduction

Automatic music composition is an important topic of computational creativity. In the last decades, a large number of generative music systems that exploit Artificial Intelligence techniques such as, machine learning, CSP and evolutionary algorithms, have been designed and developed (Fernández and Vico 2013).

In this paper, we propose an AI planning approach which automatically creates 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 (Koller and Hoffmann 2010), interactive storytelling and narrative planning (Riedl and Young 2010; Por-

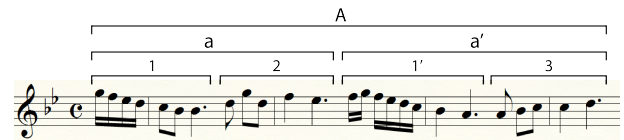


Figure 1: Example of hierarchical structure in the first movement of the Sonata KV333 by Mozart. The top level sentence is divided into two double phrases, and four phrases.

teous et al. 2011; Haslum 2012). Given the number of available high-performance planning engines and the expressivity of PDDL, 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 (Velardo and Vallati 2014), 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 firstly provide the necessary musical background. Then the flattening and the tree techniques are described and their results are discussed. Finally, conclusion are given.

Musical Background

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 (Nettl 2000). Definitions range from the idea of music as a subjective experience (Berio et al. 1985) and as organised sound (Varèse and Wen-Chung 1966), to the idea of music as a language (Cooke 1959), or as a social construct (Nattiez 1990). How-

ever, there are specific aspects of music, that although they are not sufficient to define it, would be at least necessary for an operational definition. *Structure* is one of them. Specifically, music is based on hierarchical structures (Lerdahl and Jackendoff 1985). 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 (Paas, Renkl, and Sweller 2003). 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 items only, for not more than 5 seconds (Miller 1956). 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 (Brown and Jordania 2013). This seems to imply that hierarchical organisation of music is embedded at a biological-cognitive level.

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 (Knösche et al. 2005), to detect similarities between them, and to recognise new passages as well (Krumhansl 1996). 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 (Strang and Schönberg 1967). 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* (Velardo and Vallati 2014). The former consists of filling the structures provided by the latter, with raw musical materials. In other words, the top-

```
(:action IS_EQUAL_TO
:parameters (?x ?y - phrases)
:precondition (and (is_before ?y ?x)
  (processed ?y)
  (clear ?x))
:effect (and (processed ?x)
  (increase (total-cost) 2)))
```

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

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.

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.

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.

Flattening the structure

This approach allows to encode the musical structure generation process as one planning problem, in which only musical phrases are considered. Clearly, such an approach does

```

(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)

```

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

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 $P = \langle p_1, p_2, \dots, p_n \rangle$ 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 (Strang and Schönberg 1967): (i) introducing a new phrase, (ii) 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:

A, B, B, B', C, B'', B', A, A, D

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.

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 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 4) 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

```

(: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)))

```

Figure 4: The operator used for processing two double phrases from a sentence ?sy which is similar to ?sx .

a double phrase. The `clear_similar` predicate indicates that a double phrase is a constituent of a sentence ?sx that is similar to ?sy.

Results

The approaches depicted in the previous section have been used for composing melodies. In particular, structures have been generated by using the well-known planner LPG (Gerevini, Saetti, and Serina 2003) on problems from the two approaches. 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 (Velardo and Vallati 2014). 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 folksongs of the well-known Essen database.¹

We used both flattening and tree approaches for generating the structure of a number of short melodies – approximately 40 phrases, which corresponds to 70-90 seconds of music –. The generation process took about a CPU-time second each. In order to evaluate the quality of structures, we asked 3 music experts to assess the structure of short melodies generated both with flattening and tree approaches, filling a questionnaire with quantitative and qualitative questions. To put evaluation in context, they then compared these melodies with those generated in (Velardo and Vallati 2014),

which exploited a random-walk top-down approach for the structure. The reliability of agreement between experts has been assessed through Fleiss’ kappa test (Fleiss 1971), with $k = 0.43$, which indicates a moderate agreement (Landis and Koch 1977). 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 walk. The structure of melodies composed with the tree approach and that of the Velardo and Vallati (2014) 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 folksongs”.

Conclusion

Automatic generation of music is a new and 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. 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. As expected, music experts found that the tree approach is quite effective for generating human-like melodies, while the flattened method is not.

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. 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.

¹<http://www.esac-data.org/>

References

- Berio, L.; Dalmonte, R.; Varga, B. A.; and Osmond-Smith, D. 1985. *Two interviews*. M. Boyars.
- Brown, S., and Jordania, J. 2013. Universals in the worlds musics. *Psychology of Music* 41(2):229–248.
- Cooke, D. 1959. The language of music.
- Fernández, J. D., and Vico, F. J. 2013. AI methods in algorithmic composition: A comprehensive survey. *J. Artif. Intell. Res. (JAIR)* 48:513–582.
- Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.
- Gerevini, A.; Saetti, A.; and Serina, I. 2003. Planning through stochastic local search and temporal action graphs in LPG. *J. Artif. Intell. Res. (JAIR)* 20:239–290.
- Haslum, P. 2012. Narrative planning: Compilations to classical planning. *J. Artif. Intell. Res. (JAIR)* 44:383–395.
- Knösche, T. R.; Neuhaus, C.; Haueisen, J.; Alter, K.; Maess, B.; Witte, O. W.; and Friederici, A. D. 2005. Perception of phrase structure in music. *Human Brain Mapping* 24(4):259–273.
- Koller, A., and Hoffmann, J. 2010. 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*, 238–241.
- Krumhansl, C. L. 1996. A perceptual analysis of mozart’s piano sonata k. 282: Segmentation, tension, and musical ideas. *Music Perception* 401–432.
- Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *biometrics* 159–174.
- Lerdahl, F., and Jackendoff, R. 1985. *A generative theory of tonal music*. MIT press.
- Miller, G. A. 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review* 63(2):81.
- Nattiez, J.-J. 1990. *Music and discourse: Toward a semiology of music*. Princeton University Press.
- Nettl, B. 2000. An ethnomusicologist contemplates universals in musical sound and musical culture. *The origins of music* 463–472.
- Paas, F.; Renkl, A.; and Sweller, J. 2003. Cognitive load theory and instructional design: Recent developments. *Educational psychologist* 38(1):1–4.
- Porteous, J.; Teutenberg, J.; Charles, F.; and Cavazza, M. 2011. Interactive storytelling with temporal planning. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, 1323–1324. International Foundation for Autonomous Agents and Multiagent Systems.
- Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *J. Artif. Intell. Res. (JAIR)* 39:217–268.
- Strang, G., and Schönberg, A. 1967. *Fundamentals of musical composition*. Faber & Faber.
- Varèse, E., and Wen-Chung, C. 1966. The liberation of sound. *Perspectives of new music* 11–19.
- Velardo, V., and Vallati, M. 2014. Automatic melody composition and evolution: a cognitive-based approach. In *Proceedings of the 9th Conference on Interdisciplinary Musicology CIM*.