A Musical Composition Application Based on a Multiagent System to Assist Novel Composers

Maria Navarro
Computer Science Department
Salamanca University
Pza Merced, Salamanca 37005 Spain
mar90ali94@usal.es

Juan Manuel Corchado
Computer Science Department
Salamanca University
Pza Merced, Salamanca 37005 Spain
corchado@usal.es

Yves Demazeau
Laboratoire d'Informatique de Grenoble
110 avenue de la Chimie
Domaine Universitaire de Saint-Martin-dHres
BP 53 - 38041 Grenoble cedex 9 France
Yves.Demazeau@imag.fr

Abstract

This paper presents a solution to help new composers make harmonies. A multiagent approach based on virtual organizations has been used to construct this application. This model is built by using a multiagent system. This study presents a Multi-Agent System (MAS) built with PANGEA, a platform to develop different multiagent systems, capable of composing music following the HS algorithm. The results show the success of this application in correctly composing a classical harmony.

Introduction

Interest in computational creativity has been increasing in the scientific community. Although this interest is recent, there are a number of algorithms, schemas and procedures to develop an intelligent machine, capable of creating new ideas or new artistic compositions.

Many music students, or even musicians, have problems composing or improvising melodies with their own instrument. They may find it difficult to practice their improvisation or to compose their own melody because they usually need to work with other musicians who are too busy to collaborate with them. This system was designed to assist these music students in improving their abilities.

The goal of the system is to show that a simple and general agent framework such as PANGEA (Platform for Automatic coNstruction of orGanizations of intElligents Agents) (Zato and others 2012) can build a proper and scalable music composition system. A multiagent system based on virtual organizations is used because it permits making changes in the problem specification, and can modify the music style or add new rules without altering the structural composition. Only the agents behavior needs modification. The BDI architecture was chosen for these reasons.

We will evaluate the results by considering two types of criteria. First, we will consider mathematical criteria, which include an optimization function to minimize. The smaller the value in this function for one chord, the better the chord. This function considers constraint rules that evaluate the chord obtained. These rules and the evaluation method will be detailed in Section 3.

In Western music, dissonance is the quality of sounds that seem unstable and have a need to resolve to a stable sound called consonance. The definition of dissonance is culturally conditioned, which is why a classical and an occidental music culture is considered for the evaluation of consonance. According to this criterion, we can consider these consonance intervals (in order of consonance):

- Octaves
- Perfect fourths and perfect fifths
- Major thirds and minor sixths
- Minor thirds and major sixths

We will also evaluate whether the system helps composers to make their melodies or to improvise a melody by just listening to the harmonies. This evaluation consists of evaluating the system with a number from 1 to 10.

The second section contains a brief review of algorithms in music composition, Multiagent Systems and basic concepts of virtual organizations. The third section presents our model, and our particular solution, attempting to solve the problem of harmony composition with an unknown melody, and how Virtual Organizations (VO) can help to improve this system. The last section shows some results of the system, and proposes new lines of improvement.

Background

This section presents general information about composition algorithms, concepts about MAS and VO and a brief explanation about the background of agents.

Review in composition algorithms

While grammar-based systems were initially widely used in composition tasks, today there are many other algorithms attempting to compose music. Some of these are called live algorithms (Bown 2011).

One of the most successful algorithms involves Markov models (Eigenfeldt and Pasquier 2013).There are also algorithms that uses lyrics as a variable into their compositions, as for example (Monteith, Martinez, and Ventura 2012). One interesting and notable study is that of F. Pachet (Pachet 2003).

(Hoover, Szerlip, and Stanley 2011) focused on evolving a single monophonic accompaniment for a multipart MIDI by using a compositional pattern producing network (CPPN), a special type of artificial neural network (ANN).

Agents and creativity are two disciplines that have interacted in several case studies (Martin, Jin, and Bown 2011; Lacomme, Demazeau, and Dugdale 2010).

Harmony Search Algorithm Algorithm Music improvisation aims to produce an ideal state determined by aesthetic parameters, i. e., consonance or sound balance.

The procedure has five steps described here (Geem and Choi 2007). First, it is necessary to choose the optimization function and to consider a "memory" called Harmony Memory (HM, a matrix filled with as many generated solution vectors as HMS (harmony memory size)). The new harmony is generated by a random selection, a memory consideration, by using HM and a pitch adjustment (Geem and Choi 2007). The choice of one or another is conditioned by two probabilistic parameters: PAR (Pitch Adjustment Rate) and HMCR (Harmony memory Considering Rate).

Although the new harmony is built, the constraint rules that evaluate the obtained chord must also be taken into account. For this, a threshold is established. If the chord exceeds this value, it is dismissed, and the process starts again with a new chord that replaces the rejected chord.

Finally, if the new harmony vector x has a better value for the fitness function than the worst harmony in the HM, the new harmony is included in the HM. This process is repeated over and over until the stopping criterion (maximum number of improvisations) is reached.

Virtual Organizations

In the initial development of multiagent systems, the agents were seen as autonomous and dynamic entities that evolve according to their own objectives, without external explicit restrictions on their behavior and communications (Demazeau and Müller 1990). In recent years, developers have directed their interest to the organizational aspects of the society of agents (Hübner et al. 2010). Thus, two descriptive levels are set: the organization and the agent. Agents are now seen as dynamic entities that evolve within organizations.

The following sections present a description of the system, as well as the algorithm, and the agent structures used to solve the problem.

Classical Harmony Composition

Modeling musical composition is difficult because musical objects do not have any pre-assigned connotation. That means there are as many definitions of the same object as there are belief systems in musical history. For this reason, our efforts were centered on composing music from the classical period. In this period, there were many rules for composing classical music. In particular, the following main norms are considered.

- **R1** - 8th and 5th parallels: these are produced when the interval between the i-note and the j-note of the chord n and the interval between the (i+1)-note and the (j+1)-note of the chord n+1 are both 5th or 8th.
- **R2** - Leading-note resolution. There is a rule that requires a resolution of the leading-note in the tonic.
- **R3** - Voices crossing. An ideal harmony must avoid voice i getting above voice j, when j=i+1.
- **R4** - Movements between *tension*. Each chord has a peculiar role that produces stability or instability, depending on the functions (tonic, dominant and subdominant). It is the *tension* that permits the music to evolve in the composition. For this reason, our desire is to produce a movement between chords, to prevent the music from becoming boring. Thus, the repetition of the same function over time must be penalized in some way.
- **R5** - Avoid a large interval between two pitches in a chord. This is important because if we have a big pitch in the same chord, the connection between all pitches can break.
- **R6** - Avoid a large interval between two pitches in the same voice. This rule allows building more "cantabile" melodies, in general.

With all of these constraints and rules, the following optimization equation was built to minimize:

$$\sum_{i=1}^{N}\sum_{j=1}^{3} Rank(x_{ij}) + \sum_{i=1}^{N}\sum_{j=1}^{3} Penalty(x_{ij}) \tag{1}$$

Where:

$$Rank(x_{ij}) = iRank(x_{ij}, x_{i(j-1)} + \tag{2}$$

$$ln(Tension_i) + x_{ij} - x_{(i-1)j} \tag{3}$$

$Tension(x)$ values are considered with a discrete scale from 1 to 3, depending of the tension role. If the chord is Subdominant, the tension is 1, if it is dominant, tension is 3, and if it is tonic, tension would be 2. The values of $iRank(x)$ for a specific harmonic interval are:

- 3rd, 8th interval: Value of 1
- 6th interval: Value of 1.5
- 4th interval: Value of 2
- 5th interval: Value of 2.5
- Unisone interval: Value of 3
- 2nd, 7th interval: Value of 4

$Penalty(x)$ are shown in equations 4,5,6 and 7, keeping in mind the constraints considered previously.

$$x_{(i-1)j} \equiv SI \wedge x_{ij} \neq DO \Rightarrow Penalty(x_{ij}) = 5 \tag{4}$$

$$x_{i(j-1)} \geq x_{ij} \Rightarrow Penalty(x_{ij}) = 4 \tag{5}$$

$$Tension_{i-1} = 3 \wedge Tension_i = 1 \Rightarrow Penalty(x_{ij}) = 2 \tag{6}$$

$$x_{(i-1)j} - x_{(i-1)(j-1)} = x_{ij} - x_{i(j-1)} = 5 \vee 8 \tag{7}$$

$$\Rightarrow Penalty(x_{ij}) = 3 \tag{8}$$

The algorithm starts with an initialization of the Harmony Memory (HM) matrix that is stored in the repository. Several PAR and HMCR were also tested, and we chose the best ones: 0.3 to PAR and 0.2 to HMCR. In the next section, both the structure of MAS based on VO and its advantages will be explained.

Multiagent System Structure

Virtual organizations were used to implement and develop our model. Virtual organizations provide a certain number of roles easily replaceable by an agent, depending on the context. This allows the system to be very flexible. Besides, a methodology based on VO can provide us with a global vision of the problem, the model and the possible solutions.

To design the virtual organization it is necessary to analyze the needs and expectations of the system. The result of this analysis will be the roles of the entities involved in the proposed system. The following specific roles were found:

- Composer Role: This role creates the harmonic music following their rules to achieve a goal (desire).
- Evaluator Role: This role evaluates the result of the composer role and decides if it is good enough to present it to the user.
- Interface Role: This role allows the user to interact with the system.
- Data Supplier Role: This role is an agent that accesses and stores all or most of the information needed to manage the actions that govern this system.
- Control Role: The agents that exercise this role will have overall control of the system.

To implement the roles of the VO we chose to develop a MAS. For the composer and evaluator agents, we chose a BDI agent architecture (Corchado et al. 2004), for two reasons: firstly, it is the most common deliberative agent architectures, and one of the simplest; and secondly, this structure is perfectly adapted to our requirements. The BDI agent process involves two fundamental activities: a) determining which goals should be achieved (deliberation) and b) deciding how to reach these goals (planning). Both processes should be carried out by taking into account the limited resources of each agent.

The schema in Figure 1 shows how client agents are connected to model our problem.

To begin, the composer agent has as a goal or "desire" to minimize the value of the optimization function. To achieve this goal, it has to make some rules or "intentions" (that is, the algorithm), starting from its "beliefs" or its initial stage. As we can see, the BDI architecture is perfectly suited to the agent.

Additionally, the composer agent has as a "desire" to classify the chord made by the composer agent. To achieve this goal, it has to follow its "intentions", starting with its "beliefs". Finally, the remaining agents are given communication, coordination and representation tasks.

The system was developed on PANGEA (Zato and others 2012), which provides us with certain advantages. PANGEA is a service-oriented platform that allows the open multiagent system to take maximum advantage of the distribution of the resources. With PANGEA, we can change our musical agent in order to change the composition algorithm or behavior. We can even change an agent and replace it with a multiagent system capable of communicating to compose a new music. Second, we can change our Constraint Agent.

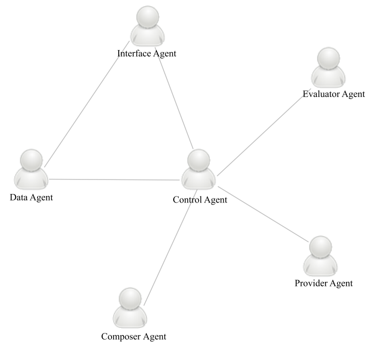

Figure 1: A global view of multiagent system interactions and communications among only client agents.

This means that different styles can be composed with this system and we only have to incorporate new behavior or update it to create jazz, rock, romantic, baroque or medieval music. We also have a database with classic styling features. The user can change these features and behaviors at any moment to permit or forbid a parallel 5th or 8th, study the leading-note resolution, etc.

Results and Conclusions

With a general framework of a MAS structure such as PANGEA, we have built a model able to compose different harmonies in order to help students new to the art of composing. However, the fitness of the results is evaluated by studying the way the rules and constraints are followed.

After the first iterations, we did not get a proper chord line, as shown in Figure 2. The first chord is perfect, taking the intervals between the notes into account. After analysing the transition between chord 1 and chord 2, we can see that the intervals are not so perfect (between Do and Re there is a 2nd interval, which is considered as dissonance). Between chord 2 and 3, the R3 is violated, as Do is becoming Mi, and the intervals again are not so perfect. Chord 4 has consonant intervals (although they might be better) but in the third voice rule R6 is violated (Sol becomes Do, and this is a little big interval.) Finally, chord 5 is better for rule R6.

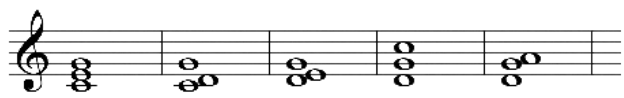

Figure 2: Harmony achieved with 45 iterations

However, the more iterations we performed, the better the results we obtained. We have a new line with 200 iterations, noticeably better than the previous one (See Figure 3). The first chord is perfect, taking the intervals between the notes into account. Analysing chord 2, we can see that the intervals are almost as perfect as chord 1 (we have a 3rd interval and a 4th interval). Chord 3 is a chord with perfect consonance. Chord 4 has a consonant 4th interval and a dissonant interval of 2nd. Finally, chord 5 is consonant with a 3rd and 4th interval.

Rules R3, R5 and R6 are respected throughout the experiment.

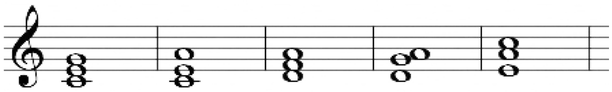

Figure 3: Harmony achieved after 200 iterations

This means that we have an evolutionary algorithm. This depends not only on the iterations we perform, but also on the parameters PAR or HMCR, which indicate the probability of making a random value for a pitch in a chord, as explained in the previous section. The fitness of the results is evaluated by studying the way the rules and constraints are followed. In other words, the more the rules are followed, the better the harmony will sound. The mathematical evaluation is to study the value of the optimization function as well as the number of the constraints that are violated.

Nevertheless in music, there is also a qualitative form to evaluate the model. This method of evaluation is based on acoustic perception, and therefore depends on the listener. We conducted tests with two experts in classical music (composers) and two non-experts in classical music to punctuate both harmonies above. The evaluation criteria was: "completely dissonant", "dissonant", "a bit consonant", "consonant", "completely consonant". The experts number 1 and number 2 evaluated the first harmony between "a bit consonant" and "dissonant", and the others evaluated as "dissonant". In the second harmony all four rated it as "consonant".

In our small study, two composers used our method and evaluated the results on a scale of 1-10. The first evaluated the result with a 6 and the second with a 7,5, which we consider as acceptable in our first approach to the system.

With regards to the virtual organization, the process of identifying and organizing roles helped to improve the management and thus to improve efficiency. The MAS structure allows us to make an extensible and scalable system as we change rules, constraints and behavior, with little effort, searching new ways of mixing different techniques, or even tools in the composition. The BDI architecture is perfectly suited for the solution we were seeking. BDI has a clear methodology that facilitates the development stage, with many theories that suit our problem. This architecture enables us to easily introduce a learning mechanism, as we can see in our case study. Moreover, using PANGEA as the platform allowed fluid communication between agents, which is evident in the design of the application, improving the modularity and the separation between client and provider as well.

As a future work, we propose incorporating rhythms. This model can also evolve to learn and self-check its own mistakes in harmony composition.

References

Bown, O. 2011. Experiments in modular design for the creative composition of live algorithms. *Computer Music Journal* 35.

Corchado, J. M.; Pavón, J.; Corchado, E. S.; and Castillo, L. F. 2004. Development of cbr-bdi agents: a tourist guide application. In *Advances in case-based reasoning*. Springer. 547–559.

Demazeau, Y., and Müller, J.-P. 1990. *Decentralized Ai*. Elsevier.

Eigenfeldt, A., and Pasquier, P. 2013. Considering vertical and horizontal context in corpus-based generative electronic dance music. In *Proceedings of the Fourth International Conference on Computational Creativity*, 72.

Geem, Z. W., and Choi, J.-Y. 2007. *Music composition using harmony search algorithm.* Springer Berlin Heidelberg. 593–600.

Hoover, A. K.; Szerlip, P. A.; and Stanley, K. O. 2011. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 387–394. ACM.

Hübner, J. F.; Boissier, O.; Kitio, R.; and Ricci, A. 2010. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems* 20(3):369–400.

Lacomme, L.; Demazeau, Y.; and Dugdale, J. 2010. Clic: an agent-based interactive and autonomous piece of art. In *Advances in Practical Applications of Agents and Multiagent Systems*. Springer. 25–34.

Martin, A.; Jin, C. T.; and Bown, O. 2011. A toolkit for designing interactive musical agents. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, 194–197. ACM.

Monteith, K.; Martinez, T.; and Ventura, D. 2012. Automatic generation of melodic accompaniments for lyrics. In *Proceedings of the International Conference on Computational Creativity*, 87–94.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32(3):333–341.

Zato, C., et al. 2012. Pangea–platform for automatic construction of organizations of intelligent agents. In *Distributed Computing and Artificial Intelligence*. Springer. 229–239.