Music Composition Using Harmony Search Algorithm

Zong Woo Geem[1] and Jeong-Yoon Choi[2]

[1] Johns Hopkins University, Environmental Planning and Management Program, 729 Fallsgrove Drive #6133, Rockville, Maryland 20850, USA
geem@jhu.edu
[2] Washington Conservatory of Music, 3920 Alton Place NW, Washington, DC 20016, USA
jycpiano@yahoo.com

Abstract. Music pieces have been composed using a behavior-inspired evolutionary algorithm, harmony search (HS). The HS algorithm mimics behaviors of music players in an improvisation process, where each player produces a pitch based on one of three operations (random selection, memory consideration, and pitch adjustment) in order to find a better state of harmony which can be translated into a solution vector in the optimization process. When HS was applied to the organum (an early form of polyphonic music) composition, it could successfully compose harmony lines based on original Gregorian chant lines.

1 Introduction

Up to now, various nature-inspired or behavior-inspired algorithms have been applied to diverse fields. Genetic algorithm (GA), one of the popular phenomenon-inspired algorithms, has also been applied to music composition.

Horner and Goldberg [1] applied a GA model to bridge-music composition in the minimalist style. The fitness function in their model was the degree of pattern match and duration. Ralley [2] proposed another GA model to develop music melody. However, the melody developed by GA could not be evaluated because there was no appropriate fitness function. Biles [3] developed an interactive GA model, GenJam, to play jazz solos. The GenJam has been applied to many jazz tunes. Currently novel research between evolutionary algorithms and music compositions has been dealt with in several workshops [4-6].

In this study, another evolutionary algorithm (harmony search or HS), inspired by music improvisation, is applied to music composition in the medieval style, where a harmony line (vox organalis) is composed to accompany a given Gregorian chant melody (vox principalis). The proposed HS algorithm was created by analogy to the music improvisation process, in which musicians improvise the pitches of their instruments to obtain better harmony [7], and has been successfully applied to various real-world applications, such as truss structure design, water network design, traffic routing, and hydrologic parameter calibration [8-11]. The HS was superior to the GA in most cases because it overcame the drawback of the building block theory of GA [12]. This study

M. Giacobini et al. (Eds.): EvoWorkshops 2007, LNCS 4448, pp. 593–600, 2007.

applies the HS to music composition problem which is to be formulated as an optimization problem with an objective function of medieval aesthetic, and the constraints of composition rules.

2 Harmony Search Algorithm

Music improvisation seeks to produce an ideal state as determined by aesthetic estimation. Similarly, algorithmic optimization seeks to produce an ideal state as determined by objective function evaluation. In the case of improvisation, the aesthetic estimation results from a set (harmony) of pitches produced by the music instruments involved, while the objective function evaluation is performed by a set (vector) of values in all the decision variables. Also, just as a harmony can be improved with each practice, likewise, the solution vector can be improved with each iteration (actually, human musician practice spans many songs, while optimization iteration spans only one song).

Figure 1 shows the analogy between music improvisation and optimization. Each musician (saxophonist, double bassist, and guitarist) is matched with each decision variable (x_1, x_2, and x_3). In addition, the range of each music instrument (saxophone = {Do, Re, Mi}; double bass = {Mi, Fa, Sol}; and guitar = {Sol, La, Si}) is matched with the range of each variable value (x_1 = {1, 2, 3}; x_2 = {3, 4, 5}; and x_3 = {5, 6, 7}), where the unit of the variables is meter if the variables stand for the pipe diameters in a water supply network.

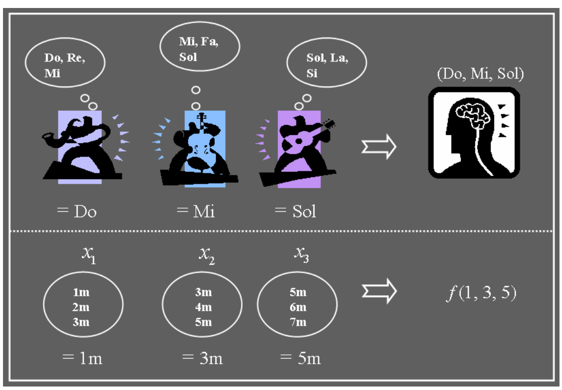

Fig. 1. Analogy between Music Improvisation and Optimization

Therefore, if the saxophonist plays the note Do, the double bassist plays Mi, and the guitarist plays Sol, their notes make a new harmony (Do, Mi, Sol). If this new harmony is "good" (aesthetically pleasing), the harmony is kept in the musician's memory. Likewise, the new solution vector (1m, 3m, 5m) generated in the optimization process is kept in computer memory if it is "good" in terms of objective function value. Just as the harmony quality is enhanced practice after practice, the solution quality is enhanced iteration by iteration. The following is an explanation of the procedure:

2.1 Problem Formulation

First, the optimization problem is specified as follows:

$$\text{Optimize } f(\boldsymbol{x}) \tag{1}$$

$$\text{Subject to } x_i \in \mathbf{X}_i,\ i = 1,2,...,N\ . \tag{2}$$

where $f(\cdot)$ is a fitness function that evaluates the fitness of improvised harmony (most evolutionary music systems employ the human user as the fitness function); $\boldsymbol{x}$ is the set of each musical instrument (decision variable) x_i; $\mathbf{X}_i$ is the set of candidate pitches for each musical instrument, that is, $\mathbf{X}_i = \{x_i(1), x_i(2),..., x_i(K)\}$ where $x_i(1) < x_i(2) < ... < x_i(K)$; and, finally, N is the number of musical instruments.

2.2 Harmony Memory Initialization

Harmony Memory (HM) matrix, as shown in Equation 3, is filled with as many randomly generated solution vectors as HMS (harmony memory size).

$$\left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_N^1 & f(\boldsymbol{x}^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(\boldsymbol{x}^2) \\ \cdots & \cdots & \cdots & \cdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} & f(\boldsymbol{x}^{HMS}) \end{array}\right] \tag{3}$$

2.3 New Harmony Improvisation

A new harmony, $\boldsymbol{x}' = (x_1', x_2',..., x_N')$ is generated by following three rules: 1) random selection, 2) memory consideration, and 3) pitch adjustment.

Random Selection. Just as a musician produces any pitch within the instrument range (for example, {Do, Re, Mi, Fa, Sol, La, Si} in Figure 2), the value of the decision variable is randomly chosen out of the value range with a probability of freedom rate (FR).

$$x_i' \leftarrow x_i' \in \boldsymbol{X}_i = \{x_i(1), x_i(2),..., x_i(K)\} \quad \text{w.p.} \quad FR \tag{4}$$

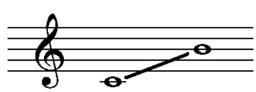

Fig. 2. Range of Musical Instrument

Memory Consideration. As a musician plays any pitch selected from the preferred pitches in his/her memory (for example, {Do, Mi, Do, Sol, Do} in Figure 3), the value of a decision variable x_i' is chosen from any pitches stored in HM ({ $x_i^1, x_i^2, \ldots, x_i^{HMS}$ }) with a probability of HMCR (harmony memory considering rate, 0 ≤ HMCR ≤ 1, HMCR = 1 – FR).

$$x_i' \leftarrow x_i' \in \{x_i^1, x_i^2, ..., x_i^{HMS}\} \quad \text{w.p.} \quad HMCR \tag{5}$$

Fig. 3. Preferred Pitches Stored in Harmony Memory

Pitch Adjustment. Once one pitch is selected based upon memory consideration, a musician can further adjust the pitch to neighboring pitches (for example, the note Sol can be adjusted to Fa or La) with a probability of HMCR × PAR (0 ≤ PAR ≤ 1) while the probability of retaining the original pitch is HMCR × (1-PAR).

$$x_i' \leftarrow \begin{cases} x_i(k \pm 1) & \text{w.p.} \quad HMCR \times PAR \\ x_i(k) & \text{w.p.} \quad HMCR \times (1 - PAR) \end{cases} \tag{6}$$

Violated Harmony Consideration. Once the new harmony $\boldsymbol{x}' = (x_1', x_2', ..., x_N')$ is obtained using the above three rules, it must then be determined to confirm to harmony rules (= problem constraints).

If the new harmony violates the constraints, it may be used, but with a penalty. For example, the rule-violating harmony of parallel fifths was nonetheless used in musical works by accomplished composers such as Bach, Beethoven, and others. It should be noted that parallel fifths were actually abundant in the context of organum, which is the example used in this study. Parallel fifths are simply an example of the application of the general model.

2.4 Harmony Memory Update

If the new harmony vector, $\boldsymbol{x}' = (x_1', x_2', ..., x_N')$ is better than the worst harmony in the HM with respect to the fitness function, the new harmony is included in the HM and the existing worst harmony is excluded from the HM. If the stopping criterion (maximum number of improvisations) is reached, computation is terminated. Otherwise, another new harmony is improvised again.

3 Medieval Music (Organum) Composition

Gregorian chant is the monophonic unaccompanied sacred song of the Roman Catholic Church in the middle ages, and organum is an early form of polyphonic music which accompanies the Gregorian chant melody. The HS algorithm is applied to the composition of organum by generating a harmony line (vox organalis) to accompany the original Gregorian chant (vox principalis).

Figure 4 shows the most ancient organum “Rex caeli Domine” contained in the anonymous book “Musica Enchiriadis” [13]. The upper line in the figure is a Gregorian chant melody and the lower line is the harmony line that was originally composed by an unknown person during the medieval era. The organum has several simple composition rules [13]: 1) the original harmony line progresses in parallel since it starts on the same note; 2) for the parallel motion, the interval of perfect fourth is frequently used while those of perfect fifth and unison (or octave) are also preferred; 3) in order to distinguish the vox principalis (chant melody) from vox organalis (harmony), the former should always be located above the latter.

Fig. 4. Score of Organum “Rex Caeli Domine”

The composition techniques for the above-mentioned organum can be formulated as the following optimization problem:

$$\text{Minimize } \sum_{i=1}^{N} Rank(x_i) + \sum_{i=1}^{N} Penalty(x_i),\ i = 1, 2, \cdots, N \tag{7}$$

$$x_i \le m_i,\ i = 1, 2, \cdots, N \tag{8}$$

$$\left|x_i - x_{i-1}\right| \le \left|m_i - m_{i-1}\right|,\ i = 2, 3, \cdots, N \tag{9}$$

$$x_{Start} = m_{Start} \tag{10}$$

$$x_{End} = m_{End} \tag{11}$$

$$x_i \in \{\text{Do}, \text{Re}, \text{Mi}, \text{Fa}, \text{Sol}, \text{La}, \text{Si}, \text{Do}^{+}\} \tag{12}$$

where x_i is the i^{th} pitch in harmony line and m_i is the i^{th} pitch in original chant line. There are 28 pitches (number of decision variables, $N = 28$) required to create a composition in this study, which represents 8^{28} (= 1.93×10^{25}) combinatorial possibilities.

Equation 7 represents the fitness function, which is the summation of the rank term and the penalty term for each pitch in the harmony line (vox organalis). As the interval of perfect fourth between vox principalis and vox organalis is most preferred, it receives the highest priority (rank = 1), as tabulated in Table 1. Table 1 shows the rankings for other intervals: a perfect fifth, unison (or octave), major or minor third (or sixth), and major or minor second (or seventh).

Table 1. Rank of Interval between Chant and Organum Pitches

Interval	Rank	Interval	Rank
Fourth	1	Fifth	2
Unison	3	Octave	3
Third	4	Sixth	4
Second	5	Seventh	5

For example, the rank for the interval of the first pitches in chant and organum lines in Figure 4 becomes three because they are unison (chant's first pitch m_1 = Do and organum's first pitch x_1 = Do), while the rank for the interval of the second pitches is five because they are major seconds (chant's second pitch m_2 = Re and organum's second pitch x_2 = Do). The lesser number (= higher rank) for the interval between chant and organum pitches is preferred because it has a smaller value in this minimization problem.

Equation 8 is the constraint that organum pitch be lower than or equal to chant pitch. If organum pitch is higher than chant pitch, a penalty value (= 5 in this study) is added to the fitness function.

Equation 9 is the constraint that the interval between two consecutive organum pitches be less than or equal to that in two consecutive chant pitches. If this constraint is violated, a penalty value (= 3 in this study) is added to the fitness function.

Equations 10 and 11 are boundary conditions that constrain starting and ending pitches in organum. Based on Gregorian chant's Latin verse, there are six notes that should be in unison (x_1 = Do, x_{11} = Re, x_{12} = Mi, x_{13} = Sol, x_{27} = Re, and x_{28} = Mi).

When applying the HS algorithm to organum composition with algorithm parameters (HMS, HMCR, and PAR are 10, 0.9, and 0.3, respectively, that are popular values in previous applications), the first improvised harmony does not appear pleasing as shown in Figure 5; it violates constraints many times (fitness value = 175).

Fig. 5. Initial Organum (Fitness = 175)

After 3,000 improvisations, the HS algorithm found the organum with a fitness measurement of 42, as shown in Figure 6. It sounds satisfactory without any awkward pitches [14].

Fig. 6. Final Organum (Fitness = 42)

The HS was further applied to the organum composition based on the more complex (N = 50) Gregorian chant "Adoro Te Devote (Godhead here in hiding)." After 10,000 improvisations, the HS algorithm found the organum with a fitness measurement of 128, as shown in Figure 7.

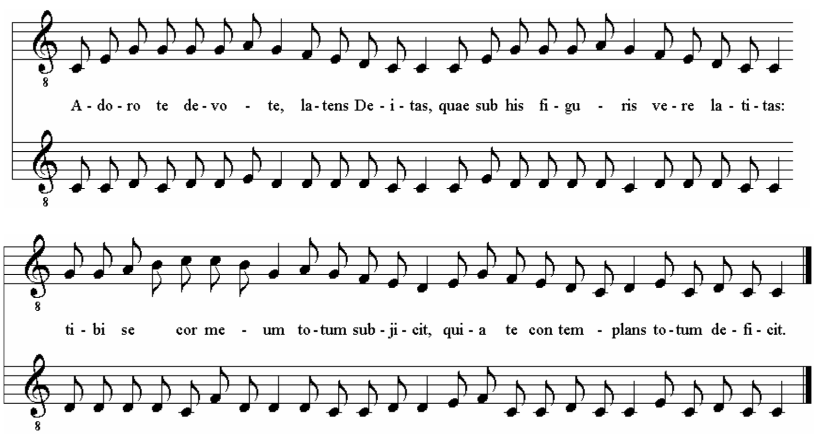

Fig. 7. Organum for Gregorian Chant "Adoro Te Devote"

4 Conclusions

A music-inspired algorithm, HS, has been applied to medieval music composition. The HS algorithm mimics the behaviors of musicians: random selection, memory consideration, pitch adjustment, and occasional violation of harmonic rules. These behaviors successfully created an organum composition in this study.

Applied to organum music composition, which was formulated as an optimization problem, HS composed satisfactory organum lines by generating up to 3,000 improvisations within one second on Intel Celeron 1.8GHz CPU. Total enumeration requires

8^{28} (= 1.93×10^{25}) function evaluations. Also, a more complex organum piece was successfully composed using the same process.

Future study of the HS algorithm should involve its application to more complex musical composition. Also, an efficient method to quantify the qualitative and subjective elements of aesthetic estimation in music composition should be developed.

References

1. Horner, A. and Goldberg, D. E.: Genetic Algorithms and Computer-Assisted Music Composition. Proceedings of the International Computer Music Conference. (1991) 437-441
2. Ralley, D.: Genetic Algorithms as a Tool for Melodic Development. Proceedings of the International Computer Music Conference. (1995) 501-502
3. Biles, J. A.: GenJam in Perspective: A Tentative Taxonomy for GA Music and Art Systems. Leonardo. 36(1) (2003) 43-45
4. Johnson, C. G. and Cardalda, J. J. R.: Genetic Algorithms in Visual Art and Music. Leonardo. 35(2) (2002) 175-184
5. Cagnoni, S., Cardalda, J., Corne, D., et al.: Lecture Notes in Computer Science (Vol. 2611) - Applications on Evolutionary Computing. Springer-Verlag, NY, USA (2003)
6. Rothlauf, F., Branke, J., Cagnoni, S., et al.: Lecture Notes in Computer Science (Vol. 3449) - Applications on Evolutionary Computing. Springer-Verlag, NY, USA (2005)
7. Geem, Z. W., Kim, J. H., and Loganathan, G. V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation. 76(2) (2001) 60-68
8. Geem, Z. W., Lee, K. S., and Tseng, C. -L.: Harmony Search for Structural Design. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). (2005) 651-652
9. Geem, Z. W.: Optimal Cost Design of Water Distribution Networks using Harmony Search. Engineering Optimization. 38(3) (2006) 259-280
10. Geem, Z. W., Tseng, C. -L., and Park, Y.: Harmony Search for Generalized Orienteering Problem: Best Touring in China. Lecture Notes in Computer Science. 3612 (2005) 741-750
11. Kim, J. H., Geem, Z. W., and Kim, E. S.: Parameter Estimation of the Nonlinear Muskingum Model using Harmony Search. Journal of the American Water Resources Association. 37(5) (2001) 1131-1138
12. Geem, Z. W.: Improved Harmony Search from Ensemble of Music Players. Lecture Notes in Artificial Intelligence. 4251 (2006) 86-93
13. Kim, M., Noh, Y., Park, M., Yi, S., Hur, Y.: Listening and Learning A History of Western Music – Volume I, SimSeolDang, Seoul, South Korea (1995)
14. http://www.hydroteq.com/