Towards an Evolution Model of Expressive Music Performance

Qijun Zhang Eduardo Reck Miranda

Interdisciplinary Centre of Computer Music Research
University of Plymouth, UK

{qijun.zhang & eduardo.miranda}@plymouth.ac.uk

Abstract – This paper presents the design of representing the performance profile with hierarchical pulse sets (i.e., hierarchical duration vs. amplitude matrices), and then applying Genetic Algorithm (GA) to evolve the hierarchical pulse sets for music interpretation, where the fitness of GA is derived from the structure of the music to be performed. In previous work [19], we have shown that GA can evolve suitable pulse sets for musical performance. Also, commonality and diversity are found among the performance profiles decided by those evolved pulse sets. This paper reports the experiment results from an improved system where a new version of fitness rules has been devised. On basis of this system, we are proposing the next steps for the research, that is, to build a dynamic model that evolves expressive music performance through agent performers' interactions.

Index Terms – Genetic Algorithm, expressive music performance, pulse set, musical structure

I. INTRODUCTION

Music performances with proper expressions are defined as expressive music performances. In the context of Western tonal music, there is a commonly agreed notion that expression is delivered in a music performance by delicate deviations of the notated musical score. Therefore, expressive music performance research is aimed at establishing why, where and how these deviations take place in a piece of music.

To build computational models of expressive performance, is to connect the properties of a musical score and performance context with the physical parameters of a performance, such as timing, loudness, tempo, articulation and so on. These models help us to gain a better understanding of expressive music performance and provide technology to implement systems to perform music. Different strategies have been employed in expressive performance research (e.g., analysis-by-measurement, analysis-by-synthesis, machine learning and so on) in order to capture common performance principles. Comprehensive reviews about these works can be found in [8, 13, 17].

As a matter of fact, social factors, including the influence of historical practices and the interactions between performers and audience, play an important role in music performance [6]. However, the frequently used strategies can help little to investigate this aspect. Therefore, the aim of our research is to build an evolutionary simulation model that takes into account these social factors by simulating the interactions among performers and listeners, through which expressive music performance profiles are believed to emerge as a result of musical constraints and social pressure.

The outline of this paper is as follows. In next section, we firstly introduce the notion of pulse set, which is used to decide the performance profile for a piece of music, and then comes the fitness of pulse sets, followed by the evolutionary procedure used in current system. We then give a demonstration on the experiments, and at last the future work.

II. MUSICAL PERFORMANCE WITH HIERARCHICAL PULSE SET

In this section we introduce the notion of pulse sets, and how we use them as performance profiles to perform musical pieces.

A. Notion of pulse set

Figure 1.a shows a pulse represented as a curve of measurements of finger pressure on a pressure sensor pad. The information in a pulse is a wrap of specific temporal patterns with amplitude patterns, and can be quantified as real numbers (width and height correspond to duration and amplitude, respectively), as depicted in Figure 1.b. A pulse can operate at different levels of temporal organization and can be grouped into a hierarchical structure [5]. Manfred Clynes proposed to represent a hierarchical pulse set as a matrix of duration and amplitude values (shown in figure 1.c), which defines the deviations of the physical attributes of musical notes. This makes it possible to generate computer performance, by modulating the physical attributes of musical notes according to these deviations.

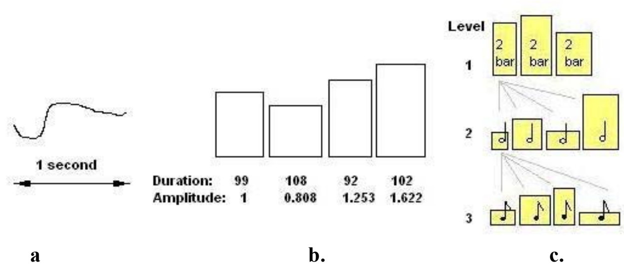

Figure 1. Illustration of a pulse and the notion of hierarchical pulse sets. (a) A pulse represented as finger pressure measurements in time. (b) A representation of pulse as a wrap of real numbers (duration vs. amplitude). (c) A hierarchical pulse set derived from grouping pulses.

B. Pulse sets as performance profiles

1) Representation of a pulse set

We adopted the notion of hierarchical pulse sets to represent performance profiles in our work. TABLE 1 shows an example of a hierarchical pulse set and its components' meanings. This example is the quantification of the pulse set drawn in Figure

1.c. To explain briefly, all the elements in a level consist of an element in its upper level. Please refer to [19] for more detailed explanation.

TABLE 1.
REPRESENTATION OF PULSE SET AND EXPLANATION

PULSE SET EXAMPLE	MEANING
8	The length of note at the lowest level {4, 8,16}={fourth, eighth, sixteenth note}
4 4 3	Number of elements in each level (From the lowest level to the highest)
0.539 0.762 0.953 1.119	Level 3 Amplitude (lowest level)
73 93 106 124	Level 3 Duration
0.853 0.798 0.998 1.333	Level 2 Amplitude
92 103 114 118	Level 2 Duration
1.398 1.476 1.464	Level 1 Amplitude
109 121 90	Level 1 Duration

2) Calculating a deviation pattern from a pulse set

The musical pieces that were used to test our system were originally stored in numerical form, as illustrated in Figure 2, together with the score it represents. For later comparing purpose, we always generate flat MIDI files based on those scores, which doesn't have any timing deviation (i.e., the rhythm is exactly as written on the score) and with even (i.e., equal) loudness for all notes.

```
2/4
-1:1.5    B3     8
0:0       E4     8
0:0.5     D#4    16
0:0.75    E4     16
0:1       F#4    4+16
1:0.25    G#4    16
1:0.5     G#4    16
1:0.75    F#4    16
1:1       G#4    4+16
......
```

Figure 2. Representation of a music piece

The pulse set example in TABLE 1 defines a performance profile containing 48 (4×4×3) pulse elements that compose the deviation pattern. The duration and amplitude, (a ratio divided by normalized value) is calculated for each element in a top down manner, by multiplying the parameters of corresponding elements in different hierarchical levels. For instance, TABLE 2. illustrates the process to decide the 1st and the 40th pulse element (represented as e1, e40) respectively as:

e1: the 1st in Level 1, the 1st in Level 2, the 1st in Level 3
e40: the 3rd in Level 1, the 2nd in Level 2, the 4th in Level 3

TABLE 2.
CALCULATION FOR A PULSE ELEMENT IN A PULSE SET.

NOTE	DURATION	AMPLITUDE
e1	$73 \times 92 \times 109 / 100^3$	1.398×0.853×0.539
e40	$90 \times 103 \times 124 / 100^3$	1.464×0.798×1.119

With this method, we can draw deviation patterns for both duration and amplitude values. Once started, these patterns repeat until the piece finishes.

3) Implementing the notes' physical parameters

We have explained in above section how to transfer a pulse set into a performance profile. In order to interpret a music piece, we need to associate the performance profile with physical parameters of all the notes in a piece, which is mainly referred to their timing and loudness in this case. This proceeds as follows: firstly, we look up a note's start time T_n in the aforementioned deviation list to infer its position along with its detailed duration and amplitude. Inspired by a method proposed by Manfred Clynes [4], a note's playing time D_n (if this note is longer than the smallest unit) is given by summing all the durations of the pulse components (100*duration value of that element), while the amplitude A_n is defined by the amplitude information of its first pulse component. Precisely speaking, T_n is the tick position which is exactly used in rendering a MIDI noteOn event. We assign each note a "note-on velocity" MIDI code value to modify its amplitude. Each velocity value is normalized within a range [L_{min}, L_{max}]. Then the position to render noteOff event is not hard to deduce with a known D_n. Finally we change this MIDI file's play back tempo by comparing its total tick length with that of the flat MIDI file, in order to get pieces of the same length.

Through the above modification, the system can produce a new MIDI file added with expressions. The next section will introduce how to evaluate the performance principles.

III FITNESS FUNCTION BASED ON MUSICAL STRUCTURE

It is commonly agreed that there is a strong relation between expression in music performance and music structure [3,7,12, 14,15,18]. This helps to explain a performer's stable performances for the same piece through years, or the existence of commonalities in the performances by different performers. On the other hand, the diversity in performance is also interesting and necessary the modelling of expressive performance is carried on.

Starting with the musical score, the approach we take is to design flexible performance principles and let the system itself to combine them in flexible ways. In other words, evolution mechanism is employed in order to lead the flexibility going towards a reliable direction.

For this purpose, we have devised descriptive performance principles without quantified regulations, which are about to be introduced in following paragraphs. And then we employ GA, whose fitness function is informed by these principles, to select and evolve suitable pulse sets, starting from randomly initiated individuals. GA is chosen here since otherwise it would be hard to design manually a decent performance profile based on simply descriptive principles. Furthermore, for the same piece, we can try with GA to evolve a number of suitable but different pulse sets. As mentioned above, this diversity is a noticeable phenomenon in real performances, and also a prerequisite for the next stage of our research.

A. Selected musical structures

Music performance mainly functions to communicate the music idea of a piece with listeners, by performers. Alike other types of human communication by acoustical signals, e.g. speech, grouping and accentuation are regarded as the most important principles to facilitate the communication. Thus we give our preference of to those performance profiles that properly highlight grouping and accentuation structure in music.

In terms of tonal music, three components — rhythm, melody and harmony have all been taken into account, as they work together towards forming the framework of a piece's structural features. To break down the problem, we analyse the development of each of them in the piece, and combine the cues from these three aspects that are related to grouping and accentuation. The resulting analysis contains two parts, timing deviation and amplitude deviation, since these are the parameters that a pulse set determines, and consequently which the evaluation is based on. As a plus, those analysis is detailed into note level, compatible with the deviation pattern decided by a pulse set. The detailed method to conclude this analysis is explained as follows.

B. Structure Analysis

In the present version of our system, the structural analysis related to grouping is a modification of Local boundary detection model (LBDM) by Cambouropoulos[1]. LBDM calculates boundary strength values for each pair of consecutive notes in a melodic surface. The higher value represents the higher strength of local discontinuities. As a modification of Cambouropoulos's model, in our system, we take into account the degree of change related to time, pitch and also harmony intervals. The latter was inspired by the "melodic charge" rule devised by Friberg and Sundberg [7], but not exactly the same, since we assign distance value to each tone according to the probe tone ratings, given by Krumhansl[9]. Figure 3 gives more explanation on this.

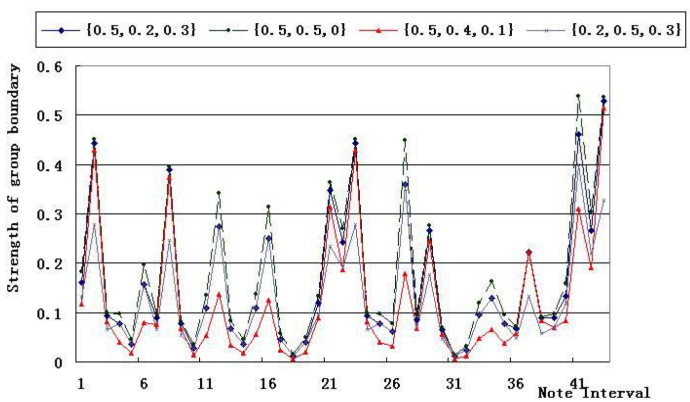

Figure 3. Strength of group boundary
Calculated as a weighted sum about inter-onset-interval, pitch interval and harmonic distance. Different line plots the resultant strength when assigning different weight to each element.

In terms of accentuation pattern, we combine metrical accentuation, melody contour and chord progression. Inspired from multi-level learning strategy by Widmer and Tobudic [18], the way to combine those analysis is to multiply corresponding values in each level, from bottom to up. The positions of chord progression is given by David Temperley's software Melism [16], and the value associated to each chord is again from Krumhansl[9]. Figure 4 illustrates this process.

C. Selected performance principles

Our descriptive performance principles are largely inspired by Eric Clarke's generative rules for expressive performance [2], which are:
Principle1: to indicate the structural direction by parametric gradients.
Principle2: to indicate group structures by parametric continuities and discontinuities.
Principle3: to accentuate individual events by local parametric (tempo/amplitude) intensification or contrast.
Another important rule we have also included is phrase arch, concerning about the tempo curve within a phrase, evident and employed by many researches [7,10,11,18]. We list it here as:
Principle4: a phrase is always performed with an initial accelerando and subsequent ritardando.

Above four rules perfectly fit our need as performance principles, considering the facts that, they all cover the features about grouping and accentuation; they are purely descriptive as no single quantity information is mentioned. Although words such as "gradients", "(dis)continuities", "contrast" are vague, they make sense in the context of the structural analysis we have got — dotted curves plotted from grouping or accentuation analysis (which we name as Curve(dur) and Curve(amp) in following text). Therefore, we concrete them with several rules that return values. And we work out the total fitness of a pulse set based on how it performs on these rules. Two components, *FitDurDev*, and *FitAmpDev* consist of the total fitness, respectively representing the fitness of timing (duration deviation) and dynamic (amplitude deviation).

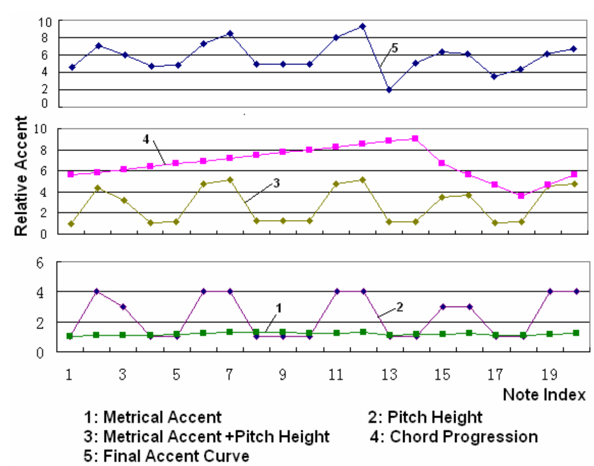

Figure 4. Accentuation pattern.
Calculated as multiplication of related elements (metrical accent, pitch height and chord progression) in corresponding positions.

1) FitDurDev

FitDurDev is obtained by a pulse set's fitness in relation to three rules. Generally speaking, as slowing down is a common expressive option when the music is approaching its group boundary, it's preferred that the note closer to group boundary is of higher positive position on Curve(dur). In addition, notes are better not to have extremely large deviation from its notated time. These three rules are:

Rule 1: For every two consecutive nodes on Curve(dur), d_m,d_{m+1}, the related notes n_d and n_{d+1} have duration deviation $durDev_m$, $durDev_{m+1}$, then a value v_1 is returned according to following calculation:

$$v_1 = \sum_{m=0}^{Nnote-2} \begin{cases} 1 & (\Delta d \bullet \Delta durDev > 0) \\ 0 & (\Delta d \bullet \Delta durDev = 0) \\ -1 & (\Delta d \bullet \Delta durDev < 0) \end{cases} *$$

$$\Delta d = d_{m+1} - d_m ,$$

$$\Delta durDev = durDev_{m+1} - durDev_m$$

Rule 2: for all local peak nodes c_p on Curve(dur), if $durDev_p < durDev_{p-1}$ and $durDev_p < durDev_{p+1}$†, v_2 is increased by 1.

Rule 3: Given a preset maximum deviation Max_{durDev}, the pulse set should be punished if a note has larger deviation than $\pm Max_{durDev}$. otherwise v_3 is increased by 1.

The value of *FitDurDev* is dependent on how well a pulse set obeys the first rule and how often the violation of the last two rules happens.

$$FitDurDev = v_1 / N_{note-1} - v_2 / N_{peak} - v_3 / N_{note}$$

N_{note} is the total number of notes in the piece and N_{peak} is the number of local boundaries.

2) FitAmpDev

The rule to decide *FitAmpDev* is similar as Rule1. It is an evaluation of how well the notes' amplitude contour fits the accentuation analysis Curve(amp).

Rule 4: For every two consecutive nodes on Curve(amp), a_m,am_{+1}, the related notes n_m and n_{m+1} have $ampDev_d$, $ampDev_{d+1}$, then a value v_4 is returned according to

$$v_4 = \sum_{m=0}^{Nnote-2} \begin{cases} 1 & (\Delta a \bullet \Delta ampDev > 0) \\ 0 & (\Delta a \bullet \Delta ampDev = 0) \\ -1 & (\Delta a \bullet \Delta ampDev < 0) \end{cases}$$

$$\Delta a = a_{m+1} - a_m ,$$

$$\Delta ampDev = ampDev_{m+1} - ampDev_m$$

3) Total fitness

In the present version of our system, we define the total fitness of a pulse set to be the weighted sum of FitDurDev and FitAmpDev. To say, *Fitness = w*FitDurDev + FitAmpDev*. We are aiming to take the privilege of this simple weighting mechanism, to differentiate listeners' higher ability on detecting small timing changes than loudness. Higher weight is associated with *FitDurDev*, we have tested both methods, either to assign a fixed value (e.g. 2) to *w* or randomly generated between 1 and 2.

* This rule is the associated effect of Principle1 and Principle4, here for Principle4, we have made the assumption for the turning point according to the strength of group boundary.

† This rule is an implementation of Principle2. Instead of setting the note at group boundary to be slower than both the notes before and after it, the choice is made here because many research has shown that it often happens that the penultimate note of a group is lengthened further than the last.

IV EVOLUTIONARY PROCEDURE

In this section we introduce the procedure to evolve suitable pulse sets from scratch.

A. Genome representation of a pulse set

A pulse set is represented by a long string of real numbers in the same order as shown in Table 1. In this string, we separate lines with ";" and insert "," between elements in the same line. This makes it convenient to access and operate on parameters of different hierarchical levels. As an example, the pulse set in Table 1 is represented as follows (for the sake of clarity, we omitted Level 2 and Level 1):

8;4,4,3; 0.539: 0.762: 0.953: 1.119;73,93,106,124; …

B. Initialization of the first generation

The individual pulse sets of the first generation are randomly generated. For the moment, we have established that all pulse sets have 3 or 4 levels. So the parameters of a pulse set that are randomly generated include:

(1) The number of levels, either 3 or 4.

(2) The length of quickest note

(3) The number of elements in each hierarchical level

(4) Amplitude and duration values for each element in every level

C. Evolution algorithm

For every generation, each pulse set is used to produce an interpretation of the given piece, as described in section 2, and a fitness value is calculated according to the definition of fitness functions introduced in section 3. Thus, we obtain an array of values $Fit0=f_1,f_2,...,f_n$, where f_i is the fitness value of the i^{th} individual pulse set. The offspring pulse sets for the next generation are created on the basis of this fitness array. The procedure is as follows:

(1) Calculate the fitness values of the current generation P0

(2) Select parent candidates to compose of population $P0_1$

(3) Operate crossover on pairs of pulse sets in $P0_1$ to get population $P0_2$ (P0 and $P0_2$ have the same population),

(4) Candidates in $P0_2$ perform mutation if applicable.

(5) Keep the best 20% of P0 and the best 80% of $P0_2$ to consist of generation P1

(6) Repeat the steps from (1) to (5) until completing a preset number of generations.

D. Genetic operations

In this section we explain the three genetic operations used in the evolution procedure: selection, mutation and crossover, respectively.

1) Selection of parents

We always choose the better half of population, namely, the individuals that have higher fitness, to be parents of next generation.

2) Mutation

There are two steps for a pulse set to perform mutation. Firstly, a randomly generated real number r (between 0 and 1) is compared with a preset mutation rate *mutRate*. If r<*mutRate*, this pulse set can perform mutation, otherwise not.

When a pulse set is performing mutation, considering its hierarchical property, we have employed three different ways for operating mutation. Figure 5 shows examples of how each of the following mutation schemes work: **Ma, Mb, and Mc.**
Ma: Randomly modify every duration or amplitude values in the pulse set. The range of changes for amplitude is [-0.1, 0.1], and for duration is [-5, 5].
Mb: Swap the order of elements in the same level of the pulse set randomly, but keep the duration and amplitude wraps.
Mc: Swap the order of hierarchical levels in the pulse set randomly.

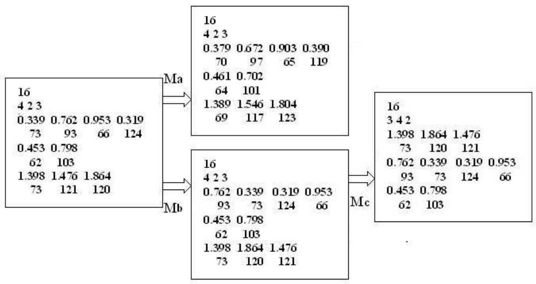

Figure 5. Examples of mutation schemes.
An integer between 1 and 3 is generated randomly for a pulse set. This number defines which mutation method to be used. 1, 2 and 3 respectively refers to Ma, Mb and Mc.

3) Crossover

To maintain the evolved parameters in hierarchical levels, we only allow for segmentation and crossover at the positions of complete hierarchical levels. Therefore, crossover enables two parents pulse sets to exchange some of their component levels. The other constraint is, the crossover points of parents have to be chosen in order to guarantee the offspring pulse sets still have 3 or 4 levels.

V. DEMONSTRATION

As a demonstration, we present an example with the melody of Robert Schumann's Träumerei. Figure 6 shows the structural analysis used for calculating the fitness value, including metrical analysis (the numbers at the bottom of the notes) and harmonic progression (with the chord names above the staves). Grouping analysis has been shown in Figure 3, where we employed the weight set {0.5,0.4,0.1} and accentuation analysis in Figure 4.

We have run several groups of experiments, each consisting of at least 30 runs of evolution procedure. In every iteration, 100 individual pulse sets are randomly generated for the first generation, and then we run 80 generations of evolution. The following data was recorded after each run: a) fitness values for each individual in a generation, including *FitDurDev, FitAmpDev* and *total Fitness*, and b) the best pulse set. If there is any change of mutation rate or fitness weight (used to calculated total fitness) in a generation or in a run, they need to be written down as well.

Our main goal in examining the results is to affirm the commonality and diversity in those evolved performances. Through listening to the interpreted music file and compare the deviation curves, we can generally tell if the structural fitness works. Besides, we also observe the genetic operators' effect, mainly refer to mutation. We will not show it in this report due to space limit.

Figure 6. The excerpt of soprano part of Robert Schumann's Träumerei.

First compared here is the duration and amplitude deviation resulted from several pulse sets, which are the best-evolved individuals through different runs, but of the same group. They have the same fitness function, which has fitness weight to be 2, so the *total fitness = 2*FitDurDev + FitAmpDev*.

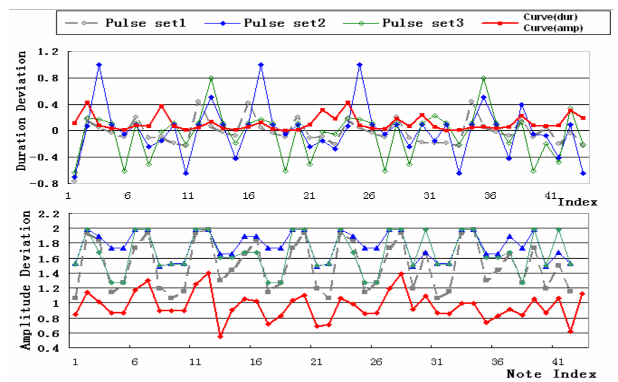

Figure 7. The evolved duration and amplitude deviation pattern. Fitness weight is fixed equal to 2.

In both of these two charts, the red line depicts the structural analysis, respectively the grouping analysis on top figure and accentuation analysis on the bottom. We also list here in TABLE 3 these three pulse sets and the fitness value they have got. It is quite straightforward to tell how a pulse set fulfil the structural fitness, by comparing the performance profile it represents with the red curve. Also, it's fair to say that those performance profiles largely share common characteristics while they are each decided by obviously different pulse sets.

TABLE 3. EXAMPLE OF "EXCELLENT" PULSE SET.

PULSE SET1	PULSE SET2	PULSE SET3
16 2 4 2 2 1.062 1.386 62 148 0.881 1.018 0.505 0.588 120 89 82 77 1.2 0.976 54 53 0.932 0.809 140 117	8 2 4 2 0.978 0.617 99 137 1.461 0.911 1.084 0.958 106 68 126 83 0.96 0.812 97 69	8 2 2 4 0.835 0.613 58 94 1.405 0.929 123 81 1.058 0.733 0.951 0.964 120 94 66 119
Fitness: 2.09	2.006	1.81

We have also tested how it works if bringing more flexibility in defining the fitness function, referring to generate different fitness weight (between 1 and 2) used in fitness function. Illustrated in Figure 8 is example performance profiles decided by three pulse sets, which are the best evolved individuals in three such runs, where each of their fitness weight is different from the other's, some even has been changing over generations. It tells that more different duration deviation patterns have been evolved, comparing with the result shown in Figure 7. This expecting result shows the feasibility to get larger number of diverse pulse sets through bringing flexible structure analysis, since the concept of fitness weight in this system represents exactly different views on giving various importance to even the same musical structure.

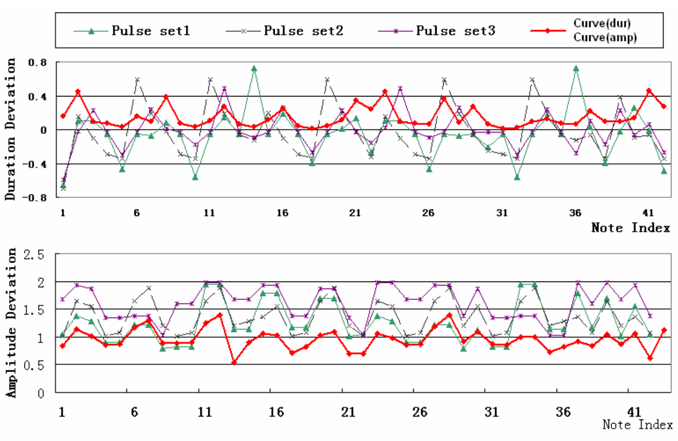

Figure 8. The evolved duration and amplitude deviation pattern. Fitness weight is randomly generated between runs or even different over generations, with value in the range of [1,2].

When listening to pieces performed with the evolved pulse sets, we can perceive the expressive dynamics of the piece. Considering the pulse sets have been evolved from random bases, we are glad to say most of the best evolved pulse sets can produce a suitable interpretation of the piece, at least in the sense of rhythmic segmentation. However, we acknowledge that such subjective assessment of the results does not hold much scientific value. We are currently developing methodology to validate the evolved pulse sets in comparison with human performance. The study of Repp B [14] is a very helpful resource for this purpose.

VI. CONCLUDING REMARKS

In this paper we introduced a novel application of GA: to evolve music performance. GA evolves suitable pulse sets for musical performance using fitness rules derived from the structure of the piece to be performed. Furthermore, the "excellent" pulse sets evolved by the GA, no matter whether they were from the same run or not, have shown diversity and also commonality. This could be observed both objectively (by comparing the figures of deviation patterns by different pulse sets) and subjectively (by listening to the "interpreted" MIDI files).

As explained earlier, we believe the dynamic characteristic of the system is largely determined by the way to combine aspects of structural analysis forming a fitness function. Thus, we plan our ongoing work to include the following:

(1) when comparing a performance profile decided by a pulse set with the aiming duration/amplitude deviation pattern, take into account not only the direction of change on the curve, but also the depth of those changes.

(2) try to design a mechanism that includes fitness weight as a parameter of each individual pulse set. This enables us moving on to the next step of the project, which is to observe the interactions among pulse sets and the effect of this on evolution.

REFERENCES

[1] Cambouropoulos E. "The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing". In Proceedings of the International Computer Music Conference (ICMC'2001) 17-22 September, Havana, Cuba.

[2] Clarke, E. F. Generative principles in music performance. In J. Sloboda (Ed.): Generative Processes in Music. The Psychology of Performance, Improvisation, and Composition. Oxford Science Publications (1988).

[3] Clarke, E. F. "Expression and communication in musical performance". (1988).

[4] Clynes, M. "Generative principles of musical thought integration of microstructure with structure". Journal For The Integrated Study Of Artificial Intelligence, Cognitive Science And Applied Epistemology 3 (1986) 185-223.

[5] Clynes, M. "Microstructural musical linguistics: composers' pulses are liked most by the best musicians". Cognition. International Journal of Cognitive Science, 55, (1995), 269-310.

[6] Davidson, J. W. and North, A. C. The Social Psychology of Music. Oxford University Press (1997).

[7] Friberg A. "Matching the rule parameters of Phrase Arch to performances of "Traumerei": a preliminary study". Proceedings of the KTH symposium of Grammars for Music Performance, May 1995

[8] Gabrielsson, A. Music performance research at the millennium. Psychology of Music, 31(2003) :221-272.

[9] Krumhansl C. Cognitive Foundations of Musical Pitch. Oxford University Press, Oxford, 1991.

[10]Neil Todd, "A computational model of rubato". Contemporary Music Review, 3, (1989), 69-88.

[11]Neil Todd, "The dynamics of dynamics: A model of musical expression". Journal of the Acoustical Society of America, 91(6) (1992).

[12] Parncutt, R 1997. "Accents and Expression in Piano Performance," in W. Auhagen et al (Eds), Systemische Musikwissenschaft. Krefeld: Dohr.

[13] Poli, G. D. "Methodologies for expressiveness modelling of and for music performance". Journal Of New Music Research 33 (2004) 189-202.

[14] Repp, B. H. "Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's Träumerei". Journal of the Acoustical Society of America (92).

[15] Sundberg, J. "Grouping and Differentiation Two Main Principles of Music", Integrated Human Brain ScienceLTheory, Method Application(Music) T.Nakad(Ed). (2000)

[16] Temperly, D. The Cognition of Basic Musical Structures. The MIT Press (2004).

[17] Widmer, G. and Goebl, W. "Computational models of expressive music performance: The state of the art". Journal of New Music Research 33 (2004) 203-216.

[18] Widmer G.,Tobudic A. "Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies" Journal of New Music Research, Volume 32, Number 3, September 2003, pp. 259-268(10)

[19]Zhang, Q, Miranda, E. "Evolving Musical Performance Profiles using Genetic Algorithms with Structural Fitness." Proceedings of Genetic and Evolutionary Computation Conference, Seattle, USA, 2006.