遺伝的プログラミングに基づく演奏生成

畑本　陽平†　　近山　隆‡

東京大学大学院 工学系研究科†　　東京大学大学院 新領域創成科学研究科‡

1　はじめに

本稿ではコンピュータの音楽への「演奏生成」への利用を取り上げる。ここでいう演奏生成とは、楽譜を解釈して個々の音の強さ、長さなどを決定し、全体的な音の並び(=演奏)をつくり上げることである。

機械的厳密に楽譜を解釈した場合、個々の音の強さや長さに寸分のずれもない演奏となる。しかし、実際の人間による演奏は決して機械的厳密ではない。コンピュータで演奏を生成する場合にも、機械的厳密から音の強さや長さをずらさなければ、その演奏は機械的で自然さを欠いたものとなる。

このような機械的厳密な演奏からのずれを生成するためにコンピュータは、そのようなずれがどのような場合に発生し、その大きさはどの程度なのかという音楽的知識を習得しておく必要がある。さらに、音楽的知識の収集のためには、人間による多数の演奏例を統計的に分析するか、機械学習の手法を用いてルールを発見させる必要がある。従来から演奏法に関する音楽的知識を収集して演奏の生成に利用した例は少なくない。現在では、コンピュータによる演奏生成システムのコンテスト[1]が開催されるに至っている。

本稿では、人間の演奏についての音楽的知識を遺伝的プログラミング(GP: Genetic Programming)によって習得し、演奏生成に利用する試みについて報告する。GP とは、プログラムをランダムに多数生成し、それらのプログラムを遺伝子に見立てて選択淘汰や変異を繰り返しながら最適解を見出す手法である。GP によって得られる最適プログラムは複雑な木構造をしていることが多いため、従来の研究で得られなかったような複雑な規則を見出せる可能性がある。

2　前提条件

これまでに音楽分野で GP が使われた例はいくつかある[2]が、演奏生成に使われた例は見当たらない。そこで、本稿で考える演奏生成は手始めとして、次のような簡単なものとする。

- 扱う音楽は単旋律であり、伴奏も和音もない。
- 鍵盤楽器、特にピアノで演奏される場合を考える。
- 演奏の生成にあたっては、個々の音の 1) 発音タイミング、2) 長さ、3) 強さをずらす。これら以外の要素については考慮しないものとする(例えば、ピッチをずらしたり、余分な音を挿入したりしない)。

Generation of Musical Performance based on Genetic Programming

† Yohei Hatamoto, Graduate School of Engineering, The University of Tokyo.
hatamoto@logos.ic.i.u-tokyo.ac.jp

‡ Takashi Chikayama, Graduate School of Frontier Sciences, The University of Tokyo.
chikayama@logos.t.u-tokyo.ac.jp

3　演奏に関する音楽的知識の習得

3.1　音楽のデータ構造

ピアノで演奏されうる多くの楽曲は、意味的にいくつかのフレーズからできていると考えられる。歌付きの曲で言えば、フレーズは歌詞の「1 行」に相当するものと考えることができる。各フレーズは 1 個以上の音符からできており、さらに音符には本来演奏されるべき機械的厳密な音の発音タイミング・長さ・強さ、それに加えて実際の演奏におけるそれらのずれ(逸脱項)が含まれる。以上の構造を図示すると図 **1**のような階層構造になる。なお、このデータ構造は、平田らの演奏生成システム[3]におけるデータ構造を簡略化したものである。

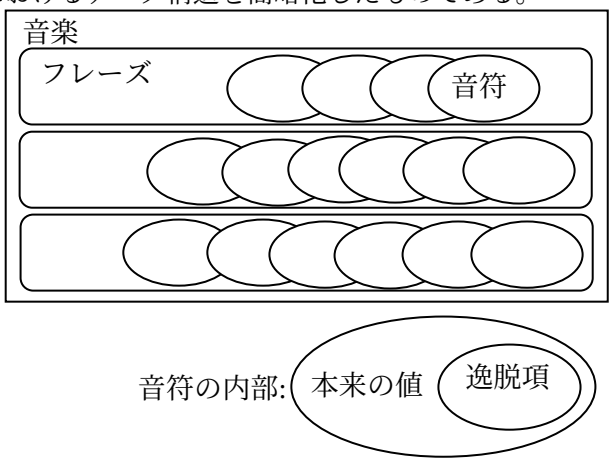

図 1: 音楽の階層的構造

3.2　GP ワークフレームの構成

前節のように構造化された音楽情報から個々の音符の逸脱項の値を求める(2で述べたように、それが演奏生成の要素となる)ための関数を得るために GP を利用する。

GP で繁殖、交雑されるプログラム(遺伝子)では、音楽を構成するすべての音符について、音の長さや前後の音の高さ、音楽の中での位置(最初、最後など)に基づいて発音タイミング、長さ、強さの逸脱の大きさを計算する。表 **1**に、遺伝子内で使われる定数・関数を列挙する。

表 1: 逸脱項計算のための定数・関数

名称	説明
(RelativePitch n)	現在注目している音符と比べて、n 個先の音符のピッチがどれだけ高いか。もしそこに音符がなければ 0 となる
(Duration n)	現在注目している音符から見て n 個先の音符の音の本来の長さ。もしそこに音符がなければ 0 となる
(RelativeTime n)	現在注目している音符の本来の鳴り始めから n 個先の音符の本来の鳴り初めまでの時間。もしそこに音符がなければ 0 となる
(CurrentPhrase o)	現在注目している音符が、初めまたは終わり(0 番目)から数えて何番目のフレーズにあるか。どちらから数えるかは、o が非負なら初めから、負なら終わりからとなる
(CurrentNoteInPhrase o)	現在注目している音符が、そのフレーズの中で初めまたは終わりから何番目にあるか。どちらから数えるかは、o が非負なら初めから、負なら終わりからとなる
(DeviationOfTime t)	発音タイミングの逸脱項の値を t に設定し、t そのものを返す
(DeviationOfDuration t)	音の長さの逸脱項の値を t に設定し、t そのものを返す
(DeviationOfVelocity v)	音の強さの逸脱項の値を v に設定し、v そのものを返す。但し、v の絶対値が 63 以上の場合、逸脱項に限り±63 に丸められる
(Add a b), (Sub a b)	a と b の和、差
(Mul a b),	a と b の積
(Div a b), (Mod a d)	a÷b の商、余り。b が 0 のときは 0 を返す
(IfEquals a b c d)	a=b ならば c、そうでなければ d
(IfGreaterThan a b c d)	a>b ならば c、そうでなければ d
(Progn2 a b)	a と b を順に評価し、b を返す
R	-250 以上 250 以下の整数定数

なお、音の高さは整数値(C4=60)、時間に関しては四分音符の長さを 960 とした相対値で表すものとした。また、音の強さに関しては、原則的に曲の中で常に一定値(=64)であるという仮定を設けた。

GP の評価関数としては、学習データとして与えた曲を構成するすべての音符について、遺伝子によって計算された逸脱と実際の人間の演奏における逸脱とのユークリッド距離を合計したものを用いる。この値を最小にする遺伝子が最も優れていることになる。

4 システムの実装と実験

筆者の提案手法の有効性を実証するため、現在これらのシステムを Java で実装し、PC 上で実験を行っているところである。次に、その手法について概要を述べる。

4.1 曲目の選定

今回実験で使用した曲目は童謡・唱歌が中心となっている。これらの曲目すべてについて、楽譜と演奏の両方のデータを作成する。また、これらの曲目は無作為に 2 分割され、一方のグループは音楽的知識の学習に、もう一方は学習した結果の正当性の評価に用いる。

4.2 楽譜データの準備

楽譜としては、独自形式のファイルを作り、これをプログラムに読ませることで図 1のようなデータ構造を生成させる。このファイルは、1 行が 1 フレーズを表し、各行はフレーズを構成する音符(音の高さと長さを書く)を空白で区切って並べたものである。

4.3 演奏データの準備

人間の演奏データとしては、筆者がキーボードで演奏し、シーケンサーで記録してスタンダード MIDI ファイルとしたものを使用する。これと楽譜データとを突き合わせ、人間の演奏における逸脱項の値を求める。

4.4 演奏に関する音楽的知識の学習

GP フレームワークでは、楽譜データから個々の音符の逸脱項の大きさを決定するための規則を求めさせる。

4.5 習得知識の評価

学習のときに使わなかった曲目の楽譜に対し、前節で得られた最も優秀な遺伝子を適用し、遺伝的プログラミングによる学習が正当であるか、過学習となっていないかを評価する。

5 おわりに

本稿では、GP によって演奏に関する音楽的知識を習得し、コンピュータによる演奏生成に利用する試みについて述べた。現段階では単旋律のピアノ演奏だけの対応となっているが、音楽のデータ表現の拡張と遺伝的プログラミングにおけるノード種の追加により、他の楽器や多声部を持つ音楽への拡張が比較的容易にできることが期待できる。

参考文献

[1] RENCON(performance RENdering piano CONcours) http://shouchan.ei.tuat.ac.jp/~rencon/index.shtml

[2] 例えば Nao Tokui & Hitoshi Iba, Music Composition with Interactive Evolutionary Computation, Congress on Evolutionary Computation 2001

[3] 平田圭二、平賀瑠美『ハーヒーフン: 2 段階演奏表情付け法によるインクリメンタルな演奏生成システム』情報処理学会研究報告 2001-MUS-39, pp19-26