

Grammatical Evolution を用いた自動作曲手法の提案

Automatic Composition using Grammatical Evolution

戸松 研人*¹ 長尾 智晴*¹

Kento Tomatsu Tomoharu Nagao

*¹横浜国立大学大学院環境情報学府

Graduate School of Environment and Information Sciences, Yokohama National University

A method of automatic composition using grammatical evolution is presented. Previously, a lot of methods of automatic composition using evolutionary computation have been presented. In this paper, we focus on the point that music have grammatical structure as with natural language, and adopt grammatical evolution (GE) that is automatic programming algorithm based on grammar, linear genome system. We defined the simple production rules in a Backus-Naur form grammar definition and fitness functions based on music theory. In experiments, we obtained interesting melodies.

1. はじめに

近年、歌声合成ソフトの開発や、動画共有サイトおよびソーシャル・ネットワーキング・サービスの発展が進んでいる。これに伴い、ユーザが自分で曲を作り、発信することへのニーズが高まっているが、音楽理論に関する専門知識や膨大な手間が必要となり、音楽の知識を持たない人にとって作曲は難しいため、現在ではコンピュータによる自動作曲技術の発展が期待されている。

自動作曲の手法として、人間が予め設定したルールにしたがって自動的に曲を構築していくルールベースの手法、既存の曲の特徴をマルコフ連鎖やニューラルネットワークによって獲得し作曲に反映する学習ベースの手法、そして、進化計算法の考え方をいっ優れた楽曲を探索する、進化計算ベースの手法が主な種類として挙げられる。本研究では、人がルールを設定する手間を省くことができ、なおかつ学習するための楽曲データベースを用意する必要のない、進化計算ベースの手法に注目する。

従来の進化計算ベースの手法として、遺伝的アルゴリズム (Genetic Algorithm; GA) を用いた手法 [Biles 94] が挙げられる。GA を用いた手法では、適切な適応度関数を設計できれば優れた解を見つけることができるため、自動作曲手法の中で一つの有力な手法となっている。しかしながら、メロディを一次元の遺伝子型で表現するため、音楽の重要な要素である、「構成」を考慮することができず、表現力に欠ける。図 1 に示すように、音楽は通常、言語と同様に階層構造を持つものであり、文法の形式を用いて音楽を説明するという試みは従来より行われている [Roads 79]。また、ジャンルや作曲家などの要素により異なる文法が用いられる。そこで、本研究では、「構成」を考慮するために、文法の要素を取り入れた進化計算法である Grammatical Evolution のアルゴリズムを用いて、「構成」を考慮した楽曲を進化的に生成することを目指す。



図 1: 音楽と「文法」.

2. Grammatical Evolution

2.1 概要

Grammatical Evolution (GE) は、予めバックス・ナウア記法 (Backus Naur Form; BNF) により定義した文法規則セットを用いて遺伝子型から表現型への変換を行うことにより、一次元配列の遺伝子から木構造を生成することができる自動プログラミング手法である。木構造の遺伝子と比較して、遺伝操作の行いやすい一次元配列を用いながら、定義した文法を適用することにより、正しい木構造の生成が可能となる。

2.2 アルゴリズム

GE のアルゴリズムを次に示す。

1. BNF による文法規則セットの定義を行う。
2. バイナリで表現された遺伝子を n -bit ずつ読み込み、10 進数列への基数変換を行う (n は任意数)。
3. 変換された 10 進数列と文法規則セットを用いて、遺伝子型から木構造へのマッピングを行う。
 - 10 進数列の最初の値を開始記号の非終端記号に対する遷移規則の数で割り、余りを求める。
 - 余りの値に応じて、遷移先を決定し、置き換える。
 - 置き換えられた記号の中で非終端記号が存在すれば、10 進数列の次の値を読み込み、開始記号と同様に遷移先を決定し、置き換える。
 - 非終端記号が存在する限り、この操作を繰り返す。10 進数列を最後まで読み込んだ場合、10 進数列の最初に戻り、値を読み込む。

連絡先: 戸松研人, 横浜国立大学大学院環境情報学府, 〒 240-8501 神奈川県横浜市保土ヶ谷区常盤台 79-7, tomatsu-kento-rc@ynu.jp

- 適応度を計算する。GA と同様に、交叉、突然変異、選択をし世代交代を行う。
- 2~4 を繰り返し、進化を進める。

3. 提案手法

3.1 文法規則セット

図 2 に今回用意した文法規則セットを示す。

```

<melody> ::= <bar><bar><bar><bar>
<bar> ::= <quarter_space><quarter_space><quarter_space><quarter_space>
<quarter_space> ::= <quarter> | <quaver><quaver>
<quarter> ::= 0.25 <MIDI_PITCH> | 0.25 REST
<quaver> ::= 0.125 <MIDI_PITCH> | 0.125 REST
<MIDI_PITCH> ::= MP60 | MP61 | MP62 | ... | MP83

```

図 2: 用意した文法規則セット。

MP60~MP83 はそれぞれ音階の C4~B5 を示す。REST は休符を示す。数値は 1 小節を 1 としたとき時の音符または休符の音価の長さを示す。<melody> の遷移先の <bar> の個数を変更することで出力される小節数を調整することができる。この文法規則セットで出力される楽曲は、楽曲小節数：8、音符の長さの最小単位：8 分音符、拍子：4 拍子に限定される。

3.2 適応度関数

今回、適応度関数として、メロディやコード進行が基本的な音楽理論に適合しているか否かを評価する関数群を用意し、それらの線形和を個体の評価値とする。関数群を次に示す。

f₁:Harmony

メロディが調やコードに対して誤りが無いかを評価する。

f₂:Smooth Melody(pitch)

音の高さに対するメロディの滑らかさを評価する。

f₃:Smooth Melody(duration)

音の長さに対するメロディの滑らかさを評価する。

f₄:Chord Adaptation

調やコード同士の繋がりの観点からコード進行の適切さを評価する。

4. 楽曲生成実験

4.1 実験概要

提案手法を用いて、楽曲生成実験を行った。楽曲小節数は 8 とし、世代数は 20000、個体数は 100 とした。

4.2 実験結果

図 3 に今回生成された楽曲譜面例を、図 4 にその楽曲の 1 小節目を表現した構文木を示す。

今回出力された楽曲は、メロディ、コード進行共に違和感の無いものであった。これは、今回用意した 4 個の適応度関数群が上手く機能したためだと考えることができる。しかしながら、全体の構成や階層構造を考慮できるような文法規則を今回は用意していないため、表現の幅が狭いことが課題として挙げられる。今後は、全体の構成や階層構造を考慮できるような文法規則を用意し、生成される楽曲に構成を持たせられるようにする必要があると考えている。



図 3: 今回生成された楽曲譜面例。

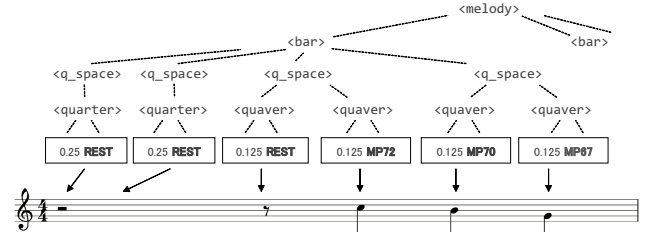


図 4: 1 小節目の構文木。

5. まとめと今後の課題

本稿では、音楽の重要な要素である「構成」を考慮するために Grammatical Evolution の考え方を導入し、自動で作曲を行う手法について提案した。文法規則セットと適応度関数を用意し、短い楽曲の生成実験を行った。結果として、違和感のない楽曲を獲得することができた。

今後の展望として、図 5 のように、進化過程でユーザ評価を取り入れ、ユーザの好みに応じ、様々な曲調の楽曲を獲得できるようにすることが挙げられる。そのために、文法規則セットを本稿で用意したものよりも表現の幅が広いものを定義する必要がある。また、ユーザ評価をどのように適応度関数に反映するかを検討し、ユーザの負担量とユーザ評価の反映性について考察する必要がある。これらの要素については口頭発表までに実験を行い、結果について考察を行う予定である。

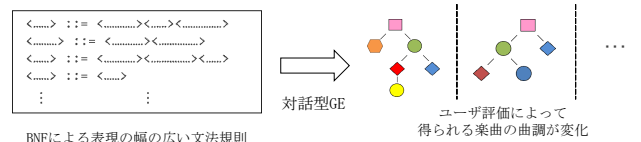


図 5: ユーザ評価の導入。

参考文献

[Biles 94] Biles, J: GenJam A Genetic Algorithm for Generating Jazz Solos, Proceedings of the International Computer Music Conference, pp.131-137(1994).

[Roads 79] Roads, Curtis and Wieneke, P: Grammers as representations for music, Computer Music Journal, Vol. 3, No. 1, pp.48-55(1979).