

## GPを用いた旋律のグルーピングルール生成

丹治 信 安藤 大地 伊庭 斉志

東京大学大学院新領域創成科学研究科基盤情報学専攻

本稿では、単旋律のグルーピングルールを自動生成することを目的とした、GP(Genetic Programming)によるルール獲得の手法について述べる。旋律のグルーピング階層の最下層グループのモチーフレベルのグループはゲシュタルト性で説明され、短期記憶の範囲で解釈されると仮定し、与えられた楽曲のグルーピングデータに対し、近傍の音符情報だけを使って、そのルールをGPで生成させた。実験では、GTTMのボトムアップのグルーピングルール4つに対し、ルールの合成を試みた。また、人間のグルーピング認知の個人差の問題があるため、主観によって作成されたグルーピングデータに対してルールの合成を試み、人間のグルーピング認知に対して考察を行う。

## Generation of melody grouping rule with GP

Makoto Tanji Daichi Ando Hitoshi Iba

Dept. of Frontier Informatics, Graduate School of Frontier Sciences,  
The University of Tokyo.

In this paper, we discuss the technique to generate a rule with GP(Genetic Programming) to aim to generate a grouping rule for monophonic melody automatically. On the assumption that motif level groups at the most lower position of group hierarchy of the melody can be explained by the gestalt, and the interpretation process is done within the range of the short-term memory, a GP grouping rule is generated for the given music using proximity notes information. The generating of the rule was tried to four bottom-up grouping rules of GTTM theory in the experiment. Moreover, the generating of the rule is tried to learn the grouping data made by human subjectivity because there is a problem of the individual variation of the group cognition. Then, human group cognition is considered.

### 1 はじめに

楽曲聴取の際、人間は様々な「音楽的構造」を認知すると言われる。特にモチーフ(動機)やフレーズと言ったいくつかの音の並びをグループと捉え、それらの階層的な構造を頭の中に構築する。また、作曲家も聞き手の認知を前提に楽曲を構成するため、これらは深く関連していると言える。

これらの楽曲構造の解析は、音楽情報処理の観点からも、コンピュータによる楽曲の理解、作曲、アレンジ、表情付け、楽曲検索等の応用あり重要である。しかし、古典的な音楽理論では言葉による人間への人間による説明がなされるのが一般的であり、それらを使って人間は楽曲を分析、解釈することが出来るが、コンピュータ上で実装するには各々の音楽用語や概念の定式化された表現が存在しなかったため困難であった。

定式化されたルールで楽曲の構造を解析しようとする音楽理論に GTTM(Generative Theory of Tonal Music) [3]がある。GTTMはゲシュタルト心理学の原理を音楽解析に応用したものであり、比較的簡潔で実装可能性の大きいルール群で記述されている。また、Mayer, Namourらが提唱するI-Rモデル[4][5]は、認知心理学の立場から旋律中の音と音との関係性を記述するもので、旋律の構造の理論を与えてくれる。

GTTMとI-Rモデルの二つのモデルに共通するのは、ボトムアップのルールにゲシュタルト心理学で言うところの近接性、類似性、連続性原理を利用し、そこから大きな構造等を構成する点にある。しかし、GTTMもI-Rモデルも基本的には人間が音楽理論等から手作業で構築したものであり、各ルールの検証の必要があると思われる。また、人間のグルーピング構造の知覚で影響があると考えられる近

接性、類似性、連続性等の要素の強さは明確に述べられていない。これらの要素は楽曲のテンポ等にも依存し [9]、また個人差があると言われている [7]。よって、これらのグループ認知には揺らぎが含まれるものと考えられる。

そこで本稿では、遺伝的プログラミング GP(Genetic Programming) によるグルーピングルールの自動獲得を試みる。GP のメリットとして、解の探索において局所解に陥りにくいことや、人間の思いもよらない解が出力される可能性等があり、それによって人間の無意識な旋律の認知処理についてのルールの生成や、その考察を行うことを目標とする。加えて、グループ認知に個人差がある場合、「その人のルール」と言うものがあるはずであり、主観によってグルーピングしたデータから、その個人差を反映したルールの獲得が出来ないか考察する。

本稿の構成は次の通りである。まず、2 章では旋律のグルーピングについての説明をする。次に 3 章と 4 章で遺伝的プログラミング GP の説明と本手法の GP の詳しい内容を説明する。5 章では 2 つの実験とその結果を報告し、6 章で結論とする。

## 2 旋律のグルーピング

本章では、本手法で前提とする人間の旋律の認知処理過程について簡単に述べる。Crowder [2] らの研究によると、人間の受け取った楽音の聴覚刺激はそのままの形 (周波数や振幅) でエコイックメモリに一時的に蓄えられる。エコイックメモリに保持されている情報は特徴抽出処理を経て一つ一つの音の情報となり、その後ピッチや先行音との特徴等からまとめられ、聴き手にグループを感じさせるものと考えられる [8]。グループには、一つの楽音、数音からなる小さなグループから、フレーズ、楽節、楽章、楽曲などのより大きなレベルがあり、小さなレベルのグループは大きなレベルのグループの構成単位となっていると考えられる。例えば、Mozart の交響曲 40 番冒頭の旋律では、譜例 1 のように階層的にグルーピングがなされる。

一般的に、階層の上の大きなグループになるほど、



図 1 譜例 1

記憶している音楽の形式や、グループ間の類似性

などが分割に影響を与え、過去の記憶や経験に基づく音楽的なスキーマが使用される。一方、階層の下のレベルのグループでは、学習や記憶による影響が少なく、多くが無意識のうちに処理されると言われる [1]。

ここでは、このレベルのグループをモチーフレベルのグループと呼ぶ。また、両者の処理過程では、階層上部のグループで短期記憶、長期記憶が処理に使われるのに対して、階層下部のグループは多くが短期記憶での無意識的な処理がなされると考えられる。これらの処理は基本的には階層の下のグループから処理されるが、上位の階層のグループによって、下位のグループの見直しが行われることもある。

本稿では、モチーフレベルのグループのボトムアップ的なルールの生成を扱う。これらは短期記憶の範囲内で処理されると仮定し、モチーフレベルのグルーピングにおいて、ゲシュタルト心理学で挙げられている近接性、類似性、連続性原理を考慮に入れ、注目している音符の近傍の情報のみを使って GP によりグルーピングルールを記述することを試みる。

## 3 GP の概要

GP(Genetic Programming) は、進化による生物の環境への適応と言う進化論的なアイデアをコンピュータ上で応用したものである。具体的には解の表現を Tree 構造で表した遺伝子型 (GType) と呼ばれるものと、この遺伝子型を持つ発現型、表現型 (PType) と呼ばれるものからなる。

環境に応じて発現型から適合度 (Fitness Value) を評価し、その適合度が高くなるように遺伝子対し進化させることで最適解を得ようとするものである。GP は大まかに次のような流れで行われる。

- ステップ 1 終端記号集合と関数記号集合を定義する。
- ステップ 2 適合度関数を定義する。
- ステップ 3 集団を初期化する。
- ステップ 4 集団の個体を全て評価し、個体に適合度を与える。
- ステップ 5 適合度をもとに交差、突然変異等の遺伝的オペレータを施し新しい集団を作成。
- ステップ 6 もし終了条件が満たされていればステップ 7 に、そうでなければ古い集団と置き換え、ステップ 4 に戻る。
- ステップ 7 最も適合度の良い個体を結果として出力する。

終了条件は問題によって様々だが、評価する世代数に上限を設ける場合や、一定数以上ベストの解が変わらなかつた場合等が条件となる。

この様に、遺伝子は交差(部分木の交換)、突然変異(新しい部分木での置き換え)等が適用され、もとの遺伝子から少しずつ変化しながら目的となる解を探索する。

## 4 グルーピングルールを生成する GP

本手法の旋律のグルーピングルールを生成する GP を説明する。まず、旋律のグルーピング問題を記述し、次にこの問題に対しての GP での記述について説明する。

### 4.1 グルーピング問題の記述

まず、旋律のグルーピング問題を次の様に表す。

**扱う旋律** 旋律は N 音符からなる時系列に並んだ単旋律とする。

**グループ** 2 つ以上の音符をモチーフレベルのグループにまとめる、その際グループは互いに重複を許さない。

**グループの表現** グループの境界を分節点と呼び、旋律の音符に分割点を与えることでグループを表現する。N 音符からなる旋律の分節点の候補は N-1 個である。

**分節のルール** グルーピングルールは、i 番目の分節候補点 が分節点かどうかを出力する関数として次のように表される。

$$G(n_{i-k}, \dots, n_{i-1}, n_{i+1}, \dots, n_{i+k}) = 0, 1$$

**階層化** 階層的なグルーピングは次の段階として、現時点では扱わない。

ここで  $n_i$  は分節候補点から見て i 番目の音符情報であり、具体的には音長 D(Duration), 占有時間 O(Occupancy Time), 後続する音との音高差 P(Pitch Interval) からなる (図 2 参照)。音高は 0 から 127

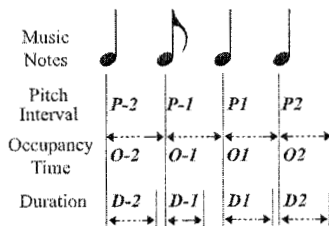


図 2 グルーピングで使用する音符情報

の整数を取るなので音高差も 0 から 127 の整数になるが、普通の旋律では一桁ほどの幅になり、特に旋律

が順次進行している時は-2 から +2 の値になる。

## 4.2 GP の問題記述

### 4.2.1 グルーピングルール生成 GP の概要

上記のグルーピング問題を GP に適用する。N 音符からなる旋律の分節候補点をインデックス 0 から N-1 まで走査し、i 番目の分節候補点 が分節点かどうかを出力するルールを GP で進化させる。(概念図を図 3 に示す)

本手法で使用した終端記号及び関数記号を表 1 に

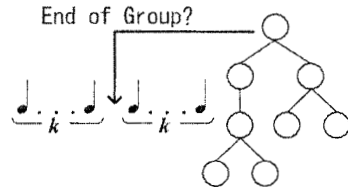


図 3 本手法の GP の概念図

示す。表において k は 5 章で後述するが、実際には 1,2,3,4 等の数が入り、どの程度まで近傍の音符を考慮に入れるかを表す数である。

個体の遺伝子は終端記号及び関数記号をランダムに使って生成され、それは Lisp の S 式の形で表されるので関数と見ることが出来る。この関数の出力を前述の関数 G と見て評価を行う。

### 4.2.2 適合度関数

旋律の分節候補点 1 から N-1 までを評価した GP の出力を  $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_{N-1}\}$  とし、予め用意した正解データを  $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_{N-1}\}$  とする。ここでデータ  $x_i, y_i$  は (0,1) の 2 値の値であり、1 の場合は i 番目の音符の後にグループの境界があることを示す。適合度関数はこれらを使って次の様に定義した。

$$F(\mathbf{X}, \mathbf{Y}) = \left( N-1 - \sum_{i=1}^{N-1} (c_1 \delta(x_i) \delta(y_i) + c_2 \delta(1-x_i) \delta(1-y_i)) + \max(\text{treeSize} - 100, 0) \right) / (N-1)$$

ただし、 $c_1, c_2$  は定数でそれぞれ 0 と 1 を正しく出力した時の重みである。この適合度関数から、 $F(\mathbf{X}, \mathbf{Y})$  が 0 に近づくほど GP の個体が良いと言える。また、簡潔なルールの獲得と目指すと言う目的の観点から木構造のサイズが 100 を超える個体に対してはペナルティーを与えている。

名前	引数	説明
Pitch Interval $n$ ( $-k \leq n \leq k-1$ )	0	現在の分節点から見て $n$ 番目の音符と $n+1$ 番目の音符の音高差を返す
Duration $n$ ( $-k \leq n \leq k$ )	0	現在の分節点から見て $n$ 番目の音符の音長を返す
Occupancy Time $n$ ( $-k \leq n \leq k-1$ )	0	現在の分節点から見て $n$ 番目の音符の始まりと $n+1$ 番目の音符の開始までの時間を返す
ONE	0	常に 1 を返す
ZERO	0	常に 0 を返す
ABS	1	引数の絶対値を返す
ADDITION	2	二つの引数の和を返す
SUBTRACT	2	一つ目の引数から二つ目の引数を引いた結果を返す
MULTIPLE	2	二つの引数の乗算の結果を返す
DIVISION	2	二つの引数の除算の結果を返す
IF LESS	2	一つ目の引数より二つ目の引数が小さければ 1 を、そうでなければ 0 を返す
IF EQUAL	2	二つの引数が等しければ 1 を、そうでなければ 0 を返す

表 1 GP の終端記号と関数記号

## 5 実験

実際の楽曲のグルーピングデータを使用して GP によるグルーピングルール生成の実験を行った。実験は 2 つ行い、GTTM のグルーピングルールの結果のデータを与え GP で生成する実験 1 と、人間が主観により与えたグルーピングデータに適合させる形でルールを生成する実験 2 からなる。以下に、実験条件と結果を示す。

### 5.1 実験条件

実験 1, 実験 2 共に実験に使用した GP のパラメータは表 2 のものを使用した。

#### 5.1.1 実験 1

実験 1 では、GP による本手法がどの程度有効かを検証するために、GTTM のグルーピングルール GPR2a, GPR2b, GPR3a, GPR3d の各ルールを合成することを試みる。

個体数	2000
最大世代数	5000
交叉率	0.8
突然変異率	0.05
初期化の最大の木の深さ	5
選択方式	ルーレット方式
エリート戦略	有り、適合度が同じ場合木のサイズが小さい個体を保存

表 2 GP のパラメータ

実験の対象とする GTTM のルールについて簡単に説明する。4 つのルールは全て  $n_1, n_2, n_3, n_4$  の音符が以下の条件を満たすとき、 $n_2$  と  $n_3$  の間が分節点と判定する。

**GPR2a**  $n_2$  の終わりから  $n_3$  の始まりまでの時間が、 $n_1$  及び  $n_3$  のそれより長い。

**GPR2b**  $n_2$  の占有時間が  $n_1$  及び  $n_3$  の占有時間より長い。

**GPR2a**  $n_2$  の終わりから  $n_3$  の始まりまでの時間が、 $n_1$  及び  $n_3$  のそれより長い。

**GPR3d**  $n_2$  の占有時間が  $n_1$  及び  $n_3$  の占有時間より長い。

これらのルールの性質上、 $k$  は 2 とし、前後 2 音符のみの情報を使う。実験に使用した旋律は、J.S.Bach のカンタータ第 140 番「目覚めよと呼ぶ声あり」BWV.140 の冒頭 24 小節のメロディーラインである。楽曲データは MusicXML 形式で与え、正解のグルーピングデータはテキストファイルで与えた。

#### 5.1.2 実験 2

実験 2 では、人間が主観から与えたグルーピングデータからルールを合成することを試みる。

本手法の GP で扱う情報は分節注目点の近傍の音符だが、 $k$  を 1 から 4 の幅で条件を変えて実験した。 $k$  の値を条件を変えて実験することで、ルール生成における近傍の音符情報の重要度が分かると思われる。また、その重要度は人間のグルーピング処理における音符情報の重要度とも相関があると考えられる。

実験に使用した楽曲は、単旋律のものが望ましいため Bach の無伴奏チェロ組曲から第一組曲の Courante を使用した。グルーピングデータに関しては人間が主観により、手作業で入力したものを用意した。



GPR	S 式表現	適合度
GPR2a	$(OC_{-1})$	0
GPR2b	$(IF\_LT (MUL (IF\_LT OC_{-1} OC_1) OC_{-1}) OC_{-2})$	0
GPR3a	$(IF\_LT (IF\_LT (ABS PT_{-1}) (ABS PT_1)) (ABS (DIV PT_{-2} PT_{-1})))$	0
GPR3d	$(IF\_IT (ABS (SUB DU_{-1} OC_1)) (IF\_IT (DIV DU_1 DU_2) (ABS (DIV OC_{-2} DU_{-1}))))$	0

表 3 生成された最良個体のルールの例

## 5.2 実験結果

### 5.2.1 実験 1

実験 1 の結果を表 3 に示す。表 3 は GTTM の各 GPR に対応した GP の出力結果を表しており、最良個体の S 式表現と適合度である。この表より、GTTM の 4 つのルールは全て適合度が最良個体で 0 になっており、合成可能であることが分かる。GP の収束の程度は 1000 世代程度であり、GPR2a や GPR2b は 10 世代程で最適解が求まっている。GPR3a は数百世代から 1000 世代程で表のルールを獲得している。一番収束に時間がかかったのは GPR3d で数百世代から数千世代程の世代数を要する。参考のため、GPR3a に対する GP の収束のグラフを図 6 に示す。

獲得されたルールに対して、例えば GPR3a を人間が手作業で書く場合素直に書けば図 3 の様になると思われるが、GP で獲得された最短のルールは図 4 のものであった。図 3 のルールはノード数が 11 個なのに対し、図 4 のルールはノード数が一つ少ない 10 個のルールで実現できている。実際、GPR3a のルールに対してこのルールは正しい出力を出す。このことから、GP によって効率的なルールを生成することが可能であると思われる、簡潔なルールの獲得を目指すと言う目的から意義があると思われる。

一方、GPR3d のルールに対しては、適合度は 0

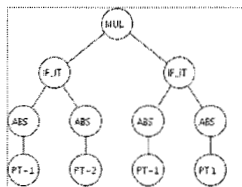


図 4 GPR3a

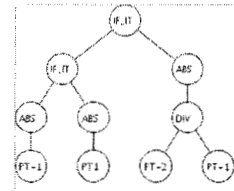


図 5 GPR3a by GP

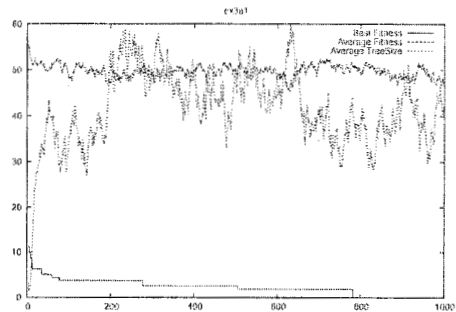


図 6 GP Fitness Graph

になったものの、実は GPR3d のルールを正しく反映してはいない。GPR3d は音長に対してのルールなので、S 式中の  $OC_i$  となっている記号が  $DU_i$  でなければならない。原因は実験で用いたデータがメロディーラインの為、ほとんど休符が入っていない為である。その場合、 $OC_i$  と  $DU_i$  が多くの場所で同じ値を返すため、どちらを選んでも適合度は変わらず、このような結果が出たと思われる。

### 5.2.2 実験 2

実験 2 の結果を図 7 に示す。図 7 は近傍の音符を 1 から 4 に変えて実験を行った結果である。実験は各条件に対して 5 回行われ、その最良個体に対する平均をとった。

結果、実験 1 とは対照的に、適合度は 0 には到達していないことが分かる。これは第一に、人間が最終的に知覚するグルーピング階層は階層間のフィードバック等があり、上位のグループから修正を受けているため、同じような音の並びでも与えたデータが違う等の可能性があることが挙げられる。また第二に、実験 1 と違い様々なルール、特に Pitch と持続時間と言う、単位の違うパラメータに対するグルーピングルールが組み合わせられているからと考えられる。

条件を変えた実験の結果、近傍の音符が 1 の場合には適合度が 0.2 ほどなのに対して、他の 3 つの場合では 0.15 程度にまで下がっている。この 3 つの

記号	頻度	記号	頻度
ABS	26	OC -4	1
ADDITION	43	OC -3	2
SUBTRACT	18	OC -2	3
MULTIPLE	13	OC -1	1
DIVISION	12	OC 1	5
IF LESS	30	OC 1	4
IF EQUAN	1	OC 2	1
DU -4	4	OC 3	1
DU -3	1	PT -4	5
DU -2	4	PT -3	11
DU -1	4	PT -2	10
DU 1	4	PT -1	27
DU 2	6	PT 1	7
DU 3	2	PT 2	5
DU 4	3	PT 3	6
ONE	1	ZERO	5

表 4 最良個体に含まれる記号の頻度

適合度には余り差が無いことから、GTTM や I-R モデルなどで扱っている、近傍の音符が前後 2 つ程度と言う数とマッチする。つまり、人間がモチーフレベルのグルーピングで考慮するのは前後 2 音符程の情報で十分である可能性が伺える。また、近傍 4 音符での実験で得られた最良個体での遺伝子型に含まれる記号を表 4 に示す。この表からも近傍-2~2 までの情報が他より平均して多いことが言える。

また、Pitch 情報が多く使われていることから、Pitch の変化に特に敏感なデータであることが言える。このように個人差を含むグルーピング知覚に置いてその差を定量的な値で示すことが出来る可能性がある。

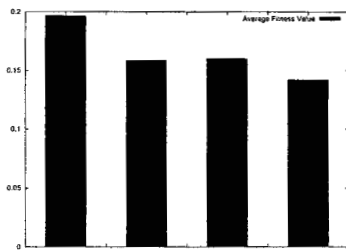


図 7 近傍数を変えたときの平均最良適合値

## 6 考察とまとめ

本稿では GP による単旋律のグルーピングルール生成を行った。今回の GP は非常に単純なモデルだが、GTTM のボトムアップのルールのような比較的

単純なルールを結果のデータから再現、生成出来ることが判った。今後、GP の効率を高めることでより複雑なルールを生成可能であると考えられる。しかし、人間が最終的に知覚するグルーピングは階層下位へのフィードバックがあると考えられるため、モチーフレベルだけのグルーピングデータが必要である。

実験 2 では、近傍の音符を 1~4 と変えて実験したが 2~4 近傍で有意差は見られず、モチーフレベルのグルーピングを行うにあたって、人間は前後 2 音符ほどの情報で処理していると考えられ、従来の音楽理論と整合するデータが得られた。

今回の実験ではモチーフレベルのグループしか扱わなかった。楽曲解析と呼ぶには、これらの結果を利用して高次のグルーピングを行う必要がある。その際、拍節情報やグループ間の類似性等、様々な要因を考慮に入れる必要があるものと思われる。

また、今回の本稿では人間の認知と言う側面を考慮して実験を行った。グルーピング構造の知覚には人間の記憶が深く関係しており、従来の音楽理論と併せてこれらの観点からも研究を進めたい。

## 参考文献

- [1] Bregman and Albert S. *Auditory Scene Analysis*. MIT Press, 1990.
- [2] Crowder and Robert G. *Auditory Memory*. Oxford University Press, 1993.
- [3] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. The University of Chicago Press, 1990.
- [4] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. The University of Chicago Press, 1990.
- [5] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. The University of Chicago Press, 1992.
- [6] Recordare. Musicxml1.1. <http://www.recordare.com/xml.html>, 2005.
- [7] 橋田光代, 野池賢二, 長田典子, 片寄晴弘. 音楽グループ構造認知に関する一考察. 情報処理学会研究報告, Vol. MUS-60, pp. 7-12, 2005.
- [8] 著: ポプ・スナイダー, 訳: 須藤貢明, 杵鞭広美. 音楽と記憶. 音楽之友社, 2003.
- [9] 福田祐一, 平田圭二, 松本裕次. 旋律中の 2 音間の時間間隔による近接性と音高差による近接性の比較について. 情報処理学会研究報告, Vol. MUS-64, pp. 47-52, 2006.