

GTTMに基づくグルーピング構造獲得システムの自動化

Automatic Parameter Tuning on GTTM-based Grouping System

岡良典*1 浜中雅俊*2*3 平田圭二*4 東条敏*1
Yoshinori Oka Masatoshi Hamanaka Keiji Hirata Satoshi Tojo

*1北陸先端科学技術大学院大学

Japan Advanced Institute of Science and Technology

*2日本学術振興会特別研究員 PD

Research Fellow of the Japan Society for the Promotion of Science

*3産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

*4NTT コミュニケーション科学基礎研究所

NTT Communication Science Laboratories

Although GTTM(Generative Theory of Tonal Music) has been considered to be one of the most promising theories in musical structure retrieval, its rules are often ambiguous and conflictive. We have proposed a grouping analyzer with adjustable parameters, by which the weight or priority of each rule could be changed. However, because those parameters were only controlled manually, it took so much a time for us to find a set of optimal parameters. In this paper, we propose an automatic parameter tuning system for the grouping rules. First, we show experimental results of random searching, hill climbing, and thereafter, show that the result of Genetic Algorithm (GA). As GA is a simultaneous multi-points searching, it enables us to find an optimal parameter set swiftly. As the crossover strategy of GA, we employ UNDX to optimize the performance. As a result, the search by GA outperforms other naïve search methods.

1. はじめに

GTTM[Lerdahl et al. 83] は 2 種類の楽曲の木構造を生成する理論である。一つはタイムスパン木と呼ばれ、楽曲中の重要な音を階層関係で示しており、もう一つは延長的還元木と呼ばれ、和声の安定性を階層関係で示している。さらにタイムスパン木の生成のために音符の集合体となるグループや、リズムの強さといったものを示した、グルーピング構造や拍節構造も定義されている。このような楽曲構造解析は、楽曲の検索や自動編曲などに活用されることが期待される。本稿ではこのうち、グルーピング構造の自動獲得について述べる。

グルーピング構造とは、調整音楽が複数のグループのつながりによって生成されており、そのグループはさらに細かく分割できるといことから得られる階層的な音のまとまりである(図1)。これまでも楽曲をグループ分割する研究は行われているが、その多くは局所的なグループ分割にとどまっている[Tmperley 01]。GTTMでは、階層的なグループを得るために複数のルールを定義しているが、その適用順序が決定されていないことや、ルールを適用する明確な基準が決められていないことなどから、ルールの競合が発生し、結果に曖昧な部分が生じる。そのことが GTTM の計算機上への実装を困難にさせている。

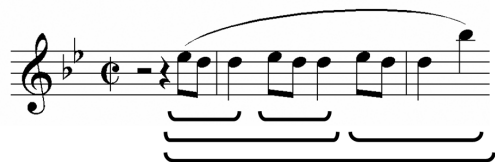


図 1: グルーピング構造

この問題を解決するために、我々はルールの強さなどを決定するパラメータを導入し、その結果として計算機上でのグルーピング構造生成を実現した [Hamanaka et al. 04]。しかしな

がら、このパラメータの調整が人手によるものに限られており、パラメータの数も多いことから、最適なパラメータを発見するためには非常に労力を要する。そこで、本稿では、自動的なパラメータ獲得について提案し、その実験結果を報告する。

2. GTTMに基づくグルーピングシステム

2.1 ルールの競合

GTTMのグルーピング構造解析には2種類のルールがあり、それぞれ、“Grouping Well-Formedness Rules”(GWFR)と“Grouping Preference Rules”(GPR)と定義されている。GWFRではグルーピングに関する制約が定義されており、GPRでは実際のグルーピングを行うルールが定義されている。よって本稿で扱うシステムでは、GPRを計算機上へ実装し、グルーピング解析を実現している。

しかし、GPRには優先順位や明確な基準が定められていないことから、ルールの競合という問題が発生する。図2は、競合の簡単な例を示しており、GPRで定義されるルールのGPR3aとGPR3dが競合していることを表している。GPR3aでは音高に大きなズレがある場合にグループの境界になりやすいことを定義しており、GPR3dでは互いに隣接する音長の変化する場所がグループの境界になりやすいことを定義している。またGPR1では、単音がグループになることは避けることが定義されていることから、この場合、音符‘2’が単独でグループになることはない。そして音符‘1-2’間か‘2-3’間のどちらにグループの境界が出来るかは、GTTMの定義のみでは知ることができない。これがルールの競合である。



図 2: ルールの競合の例

こういったケースは楽曲中の様々な場所で、また種々のル

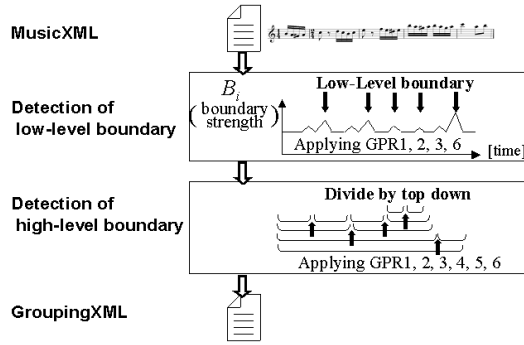


図 3: システムの流れ

表 1: 各パラメータの説明

パラメータ	意味
S^{GPRj}	各ルールの強さ． ただし， $j = (2a, 2b, 3a, 3b, 3c, 3d, 4, 5, 6)$ ．
T^{GPR4}	GPR2 と GPR3 の効果が明白な部分かどうかを決める閾値．
$T^{low-level}$	遷移 i が低いレベルでの境界かどうかを決める閾値．
σ	GPR5 で用いる正規分布の分散．
W_r	平行性に関して，リズム方向のズレと音高方向のズレのどちらを重視するか決定するパラメータ．
W_s	平行性に関して，パラレルな区間の始まりを重視するか終りを重視するかを決めるパラメータ．
W_l	平行性に関して，パラレルな区間の長さをどれくらい重視するかを決めるパラメータ．

ル間で発生するため，これまで計算機上でグルーピング構造解析を実現することは難しかった．本システムでは，ルールそれぞれを定式化し，調節可能なパラメータを用意することにより，グルーピング構造を精度よく獲得することを可能にしている．

2.2 グルーピングシステム

ここでは，本稿で扱うグルーピング構造獲得システムを紹介する．

図 3 はこのシステムの大まかな流れを示している．システムでは，楽譜を XML 化した MusicXML を入力として用いる．そして出力として GroupingXML を得ることができる．

図 4 は各パラメータと各変数の関係を示している．また表 1 は各パラメータの役割を示している．以下では，それぞれの変数の求め方と，そのときの各パラメータの働きを説明していく．

基本変数の算出 基本変数とは，消音時刻から次の発音時刻までの間隔 $rest_i$ ，発音時刻間隔 ioi_i ，音高の差 ref_i ，ダイナミクスの差 dyn_i ，楽譜上の音符の長さを実際に演奏された音の長さの比の差 $arti_i$ *1，音価の差 len_i である．ここで添字の i は遷移を表している*2．

GPR2,3,4 の適用 $D_i^{GPRj(j=2a,2b,3a,3b,3c,3d,4)}$ は，GPR のルールが成立する度合いを表す変数であり，1 が成立，0 が不成立である．GPR2,3 の 6 つのルールは各基本変数の変化が大きいところで成立する．

*1 GTTM では，アーティキュレーションの意味について明確な定義がないため，文献 [Lerdahl et al. 83] での適用例からこのように判断している．

*2 例えば $i = 1$ は第 1 音から第 2 音への遷移を表している

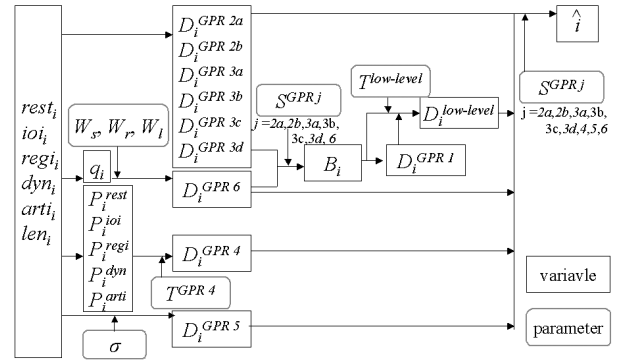


図 4: パラメータと各変数の関係

$$D_i^{GPR2a} = \begin{cases} 1 & rest_{i-1} < rest_i \text{ and } rest_i > rest_{i+1} \\ 0 & rest_{i-1} \geq rest_i \text{ or } rest_i \leq rest_{i+1} \end{cases}$$

$$D_i^{GPR2b} = \begin{cases} 1 & ioi_{i-1} < ioi_i \text{ and } ioi_i > ioi_{i+1} \\ 0 & ioi_{i-1} \geq ioi_i \text{ or } ioi_i \leq ioi_{i+1} \end{cases}$$

$$D_i^{GPR3a} = \begin{cases} 1 & regi_{i-1} < regi_i \text{ and } regi_i > regi_{i+1} \\ 0 & regi_{i-1} \geq regi_i \text{ or } regi_i \leq regi_{i+1} \end{cases}$$

$$D_i^{GPR3b} = \begin{cases} 1 & dyn_{i-1} = 0 \text{ and } dyn_i \neq 0 \text{ and } dyn_{i+1} = 0 \\ 0 & dyn_{i-1} \neq 0 \text{ or } dyn_i = 0 \text{ or } dyn_{i+1} \neq 0 \end{cases}$$

$$D_i^{GPR3c} = \begin{cases} 1 & arti_{i-1} = 0 \text{ and } arti_i \neq 0 \text{ and } arti_{i+1} = 0 \\ 0 & arti_{i-1} \neq 0 \text{ or } arti_i = 0 \text{ or } arti_{i+1} \neq 0 \end{cases}$$

$$D_i^{GPR3d} = \begin{cases} 1 & len_{i-1} = 0 \text{ and } len_i \neq 0 \text{ and } len_{i+1} = 0 \\ 0 & len_{i-1} \neq 0 \text{ or } len_i = 0 \text{ or } len_{i+1} \neq 0 \end{cases}$$

GPR4 は，GPR2,3 の効果が比較的大きなところで認識される． P_i^{rest} ， P_i^{ioi} ， P_i^{regi} ， P_i^{dyn} ， P_i^{arti} はそれぞれの基本変数に対応したルールが明白なときに大きくなる値である． T^{GPR4} は閾値を示している．

$$D_i^{GPR4} = \begin{cases} 1 & \max(P_i^{rest}, P_i^{ioi}, P_i^{regi}, P_i^{dyn}, P_i^{arti}) > T^{GPR4} \\ 0 & \text{else} \end{cases}$$

GPR5 の適用 GPR5 はグルーピングの分割が長さの等しい 2 つの部分からなるようにする．ここでは，平均をグループの中心，分散を σ とする正規分布を用いている．分散 σ は調節可能なパラメータで， σ が 0 に近いほどグループの中心付近での D_i^{GPR5} の値が大きくなり，中心に近いほどグループの境界になりやすい．逆に σ が大きくなった場合，中心付近以外でもグループの境界になりやすくなる．

$$D_i^{GPR5} = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{\left(\sum_{j=start}^i ioi_j - \sum_{j=start}^{end} ioi_j / 2 \right)^2}{2\sigma^2} \right\}$$

GPR6 の適用 GPR6 は平行性に関するルールである．ここでは 3 つのパラメータ W_r ， W_s ， W_l ($0 \leq W_r, W_s, W_l \leq 1$) を導入する (表 1 参照)．境界はこの 3 つのパラメータにより決定される．

GPR1 の適用 GPR1 の実装は，前後の遷移より局所的な境界が深いかどうかで判別する． B_i は，局所的境界の深さを 0 から 1 の実数で表したものであり，大きいほど境界が深いことを意味する． S^{GPRj} ($0 \leq S^{GPRj} \leq 1$) は各ルールの強さを決めるパラメータである．

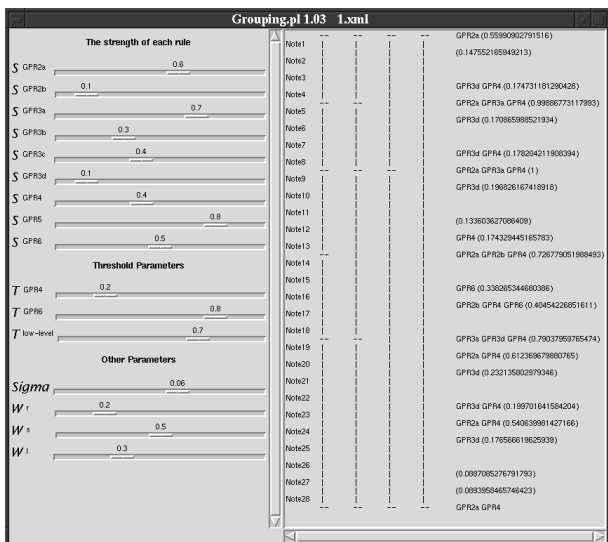


図 5: 手動によるパラメータチューニング

$$B_i = \frac{\sum D_i^{GPRj} \times S^{GPRj}}{\max_{i'} (\sum D_{i'}^{GPRj} \times S^{GPRj})}$$

局所的境界の検出 局所的境界は、 $D_i^{GPRj} (=1, 2a, 2b, 3a, 3b, 3c, 3d, 6)$ を用いて検出する。 $T^{low-level}$ は遷移 i がグループの境界になる ($D_i^{lowlevel} = 1$) かならないか ($D_i^{lowlevel} = 0$) を決定するための閾値である。

$$D_i^{lowlevel} = \begin{cases} 1 & B_i > T^{low-level} \text{ and } D_i^{GPR1} = 1 \\ 0 & B_i \leq T^{low-level} \text{ or } D_i^{GPR1} \neq 1 \end{cases}$$

階層的なグルーピング構造の検出 階層的なグルーピング構造は、ボトム処理により求めた局所的境界を用いて、トップダウンに獲得する。グループがその内部に局所的境界を含んでいる場合、次式によって、次の階層の境界が再帰的に求まる。

$$\hat{i} = \arg \max_i \left(D_i^{low-level} \times \sum_j D_i^{GPRj} \times S^{GPRj} \right)$$

評価実験 実験において、入力にはクラシック曲から 8 小節の長さを抜き出したものを用いる (ただし、モノフォニー*3である)。それに対して、グルーピングの正解データを用意する。この正解データは、1 人の音楽の専門家が作成したもので、それを GTTM の専門家 3 人によってクロスチェックしたものである。

図 5 は、このシステムのインターフェースである。左にユーザーが調節可能なパラメータ、右にそのパラメータが適用されたときのグルーピング結果が示されている。実験では、この出力結果を正解のグルーピングと比較しながらパラメータを調節していく。

システムの出力した GroupingXML に対する評価には F 値*4 を用いる。

*3 和音を含まない単旋律

*4 F 値は適合率 $P(Precision)$ と再現率 $R(Recall)$ から得られる。

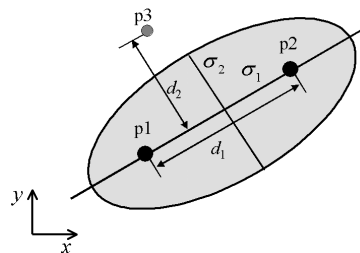


図 6: UNDX (2 次元の場合)

3. 自動化手法と実験

ここでは、パラメータチューニングの自動化手法について述べ、その実験結果を示す。我々はまず単純な手法として、ランダム探索と山登り探索によるパラメータの最適化を試みた。しかしこれらの手法では、手動より良い結果を得ることが難しいことがわかった。そこで我々は、パラメータが相互に関係を持っていることを考慮し、同時多点探索が可能な遺伝的アルゴリズム (GA) による探索を行った。

3.1 UNDX-GA

GA では、各遺伝子座を各パラメータとして扱う。そして、交叉手法として、UNDX (単峰性正規分布交叉) [Ono et al. 99] を用いる。これは UNDX が、初期の段階では幅広い空間から探索を始め、中盤から最適解がありそうな部分的な範囲を探索するため、今回のパラメータチューニングのように、解がどのような空間に存在しているか予想がつかない実験において有効ではないかと考えた。

UNDX では基本的に子は二つの親を結び線分の周辺に正規分布にしたがって生成され、第三番目の親は正規分布の標準偏差を決めるために補助的に用いられる (図 6)。子は次式によって与えられる。

$$\begin{aligned} \vec{C} &= m + z_1 e_1 + \sum_{k=2}^n z_k e_k \\ m &= (\vec{p}_1 + \vec{p}_2) / 2 \\ z_1 &\sim N(0, \sigma_1^2), \quad z_k \sim N(0, \sigma_2^2), \quad (k = 2, \dots, n) \\ \sigma_1 &= \alpha d_1, \quad \sigma_2 = \beta d_2 / \sqrt{n} \\ \vec{e}_1 &= (\vec{P}_2 - \vec{P}_1) / |\vec{P}_2 - \vec{P}_1| \\ \vec{e}_i &\perp \vec{e}_j, \quad (i \neq j), \quad (i, j = 1, \dots, n) \end{aligned}$$

ここで n は次元数、 \vec{P}_1, \vec{P}_2 は両親、 \vec{C} は子、 d_1 は両親間の距離、 d_2 は第三の親と両親を結び軸との距離、 \vec{e}_1 は両親を結び軸方向の単位ベクトル、 $z_1 \sim N(0, \sigma_1^2)$ と $z_k \sim N(0, \sigma_2^2)$ は正規乱数を表す。 α, β はユーザーが与えられる定数であるが、 $\alpha = 0.5, \beta = 0.35$ が推奨されている。また $\vec{e}_i (i = 2, \dots, n)$ は \vec{e}_1 に垂直かつ線形独立な単位ベクトルで、任意の線形独立なベクトル集合からグラムシュミットの直交化法により求められる。

生存戦略には親と子の集団から、2 個体を次の世代に残す戦略をとる。一つは最良の個体であり、もう一つはルーレット選択で決定される個体である。

ここでは、正解データのグループがシステムの出力に含まれる割合を P 、システムの出力したグループが正解データに含まれる割合を R としている。

$$F = 2 \times \frac{P \times R}{P + R}$$



図 7: 結果の例 (14. Menuet G dur)

3.2 実験結果

実験で GA に用いたパラメータは、様々な設定で数曲を解析した予備実験の結果から、集団サイズが 100、親の数は 2^{*5} 、子の数は 20 と設定した。また試行は 10 回行うことにした。

実験の終端条件は、 F 値が 1.0 になった場合、もしくは子の数 2000 に達したときとしている。これは、1 曲に対する総評価回数の上限が約 20000 回 (子の総数 \times 試行回数) であることを示している。この結果から比較のため、ランダム探索と山登り探索に対しても総評価回数の上限を 20000 回に設定した。

表 2: 各手法での F 値

Melody	manual	Random Search	Hill-Climb Search	GA (UNDX)
1.Hallelujah	0.890	0.917	1.000	1.000
2.Die Moldau	0.629	0.629	0.629	0.788
3.Ave Maria	0.349	0.388	0.527	0.636
4.Wiegenlied	1.000	0.756	0.824	1.000
5.Le coucou	0.091	0.089	0.091	0.091
6.Estudiantina	0.870	0.870	0.957	0.957
7.Sonate K.331	0.760	0.495	0.760	0.825
8.Serenate No.13	0.917	0.808	0.857	0.867
9 Air de toreador	0.833	0.857	0.923	0.952
10.Petit chien	0.583	0.583	0.689	0.697
11.Valse No.7	0.837	0.768	0.873	0.955
12.La Primavera	0.703	0.786	0.853	0.933
13.Turkischer Marsch	0.626	0.889	0.933	0.947
14.Menuet G dur	0.870	0.775	0.775	0.929
15.Blumenlied	0.923	0.895	0.947	0.947
⋮	⋮	⋮	⋮	⋮
Average	0.764	0.706	0.788	0.842

表 2 は、実験結果の一例を示している^{*6}。また、図 7 は、GA によって F 値が向上した例である。この例では、単純に一つグルーピングが増えただけだが、パラメータを手動で調節するときに、このような場合でもいくつかのパラメータを同時に変化させる必要があるため、自動化が有効であると考えられる。

100 曲の F 値を各手法で見ると、ランダム探索では 27 曲で上がっているが、49 曲で下がっており、山登り探索では、55 曲で上がっているが、24 曲で下がっている。これに対し、GA では 68 曲で上がったのに対し、下がったのは 5 曲にとどまっている。このことから、本システムのパラメータ調整には

*5 UNDX は親の数を 2 以上に設定できるが、本研究では予備実験の結果より、親の数による性能の向上が認められ無かったため、親の数が 2 である UNDX-2 を用いている。

*6 それぞれの楽曲の作曲者は、1: G.F.Händel, 2: B.Smetana, 3: F.P.Schubert, 4: J.Brahms, 5: L.-C.Daquin, 6: E.Waldteufel, 7,8: W.A.Mozart, 9: G.Bizer, 10,11: F.F.Chopin, 12: A.Vivaldi 13,14: L.V.Beethoven, 15: G.Lange.

単純な探索手法では性能を悪くする可能性が高く GA が大きな成果を発揮出来ることがわかった。

また、100 曲の最適なパラメータの値を見ると、ある一定の値におさまっているのではなく、楽曲別に最適なパラメータの値が違ってもわかっている。つまり、すべての楽曲を精度よくグルーピングするパラメータが存在しているわけではないことを示している。

GA によって F 値が上がらない楽曲については、グルーピングシステムそのものが問題となっている場合がある。これはパラメータが楽曲全体に影響するため、各所でルール of 優先度を変えられないことや、楽曲によっては平行性の認識が 1 音ずつずれて認識するといった問題である。今後グルーピングの精度をあげるにはこのような点を改良していく必要があると考えられる。

4. まとめ

我々は、グルーピング構造獲得システムにおいて、パラメータ調節の自動化を提案した。そして実験により、パラメータ調節の自動化には、単純な探索手法での最適化はあまり有効ではなく、UNDX を用いた GA により、 F 値が大幅に向上することを示した。

現在のところ最適なパラメータを見つけるには、グルーピングの正解データを用意する必要がある。今後、今回の実験で得ることのできた最適なパラメータが、正解データを必要としないグルーピングに利用できないかなど、検討していく必要があると考えられる。

今後、拍節構造解析の自動化などを実現し、最終的にタイムスパン木を自動獲得できることが目標となる。

参考文献

- [Lerdahl et al. 83] Lerdahl, F. and Jackendoff, R: A Generative Theory of Tonal Music, The MIT Press, 1983.
- [Temperley 01] Temperley, D: The Cognition of Basic Musical Structure, The MIT Press, 2001.
- [Hamanaka et al. 04] Hamanaka, M., Hirata, K., and Tojo, S: Automatic Generation of Grouping Structure based on the GTTM, Proceedings of the International Computer Music conference, 2004
- [Ono et al. 99] 小野 功, 佐藤 浩, 小林 重信: 単峰性正規分布交叉 UNDX を用いた実数値 GA による関数最適化, 人工知能学会誌, Vol.14 No.6, 1999.