

遺伝的アルゴリズムを用いたファミコン風自動編曲システムの生成

田原 花蓮^{†1,a)} 植村 あい子^{†1,b)} 北原 鉄朗^{†1,c)}

概要: 本研究は既存のポピュラー音楽をファミコン風の音楽に自動編曲することを目的としたものである。ファミコン音楽は最大同時発音数が4音までという制約が決められているため、音を削除しなければならない。そこで各パートに対して適切な音削除が行えるよう、各パートに対して音を簡略化する「変換ルール」を複数設計し、遺伝的アルゴリズム (genetic algorithm : GA) を用いて最も相応しい変換ルールをパートごとに決定していく手法を用いてファミコン風音楽の生成を行った。MIDI ファイルを用いてファミコン風アレンジを行い、主観評価と客観評価に分けて評価実験を行った。主観評価では、プロの作曲家に評価 (1 から 7 の7段階) してもらったところ、リズムの自然さについては、変換ルールありで小節ごとに GA を行ったアレンジ 1 では平均 5.8 が得られた。一方、主旋律とベース以外のパートの選び方の妥当性については、ランダムよりも低い結果になるなど、課題が残った。客観評価では、変換ルールありで小節ごとに GA を行ったアレンジ 1, 続いて変換ルールなしで小節ごと GA を行ったアレンジ 4 が総合的に高い評価が確認できた。

1. はじめに

近年、流行の曲や懐古な曲にアレンジを加え、ファミコン風の音楽にする“ファミコン風アレンジ”の動画が増え続けている。ファミコンとは、1983年に任天堂より発売された家庭用のゲーム機器、ファミリーコンピュータ (Family-Computer) の略記である。1985年に任天堂から発売されたファミコン用ソフトの「スーパーマリオブラザーズ」が大ヒットし、ゲームのみならずゲームの中で使われるBGMも大きく注目されるようになった。ゲームの中で使われるBGMが“ファミコン音楽”と呼ばれるようになり、ファミコン音楽は音楽ジャンルの一種となった。現在、様々な曲をファミコン音楽にアレンジした動画がYoutubeやニコニコ動画等の人気動画サイトに多くアップされている。人気曲をファミコン音楽風にアレンジした音源は“ファミコン風音楽”と呼ばれることが多く、ファミコン風音楽の検索ヒット数はおよそ38000件、そのなかで再生回数の多い動画は50万回再生されているものもある。この数から見るに決してマイナーなジャンルではないことがわかる。ファミコン世代に留まらず、ファミコン音楽は現在も多くの人を魅了するジャンルにある。

ファミコン音楽はファミコンに搭載されている音源チッ

プの機能が制限されているため、一度に同時に鳴らすことのできる最大同時発音数が限られている。その最大同時発音数は4音で、矩形波2音、三角波1音、ノイズ1音と音の種類も限られている。もし、初心者が自分の好きな曲をファミコン風の曲に簡単にアレンジをすることができたら、原曲を聴いた時との違った楽しみ方や魅力を発見することができる。しかし動画サイトにあるファミコン風にアレンジした曲は最大同時発音数4音という制約を破っているものがほとんどである。これは、ファミコン風にアレンジされた音楽を楽しむのだったら制約を守る必要がないと考えている可能性がある。制約を守っていないファミコン風音楽は、本来のファミコン音楽に比べて音の厚みが増しており、ファミコン音楽の特徴を十分に再現できているとは言えない。また、自分の好きな曲をファミコン風の音楽にアレンジをしようと思っても音楽経験者でないと編曲は難しい。

自動編曲についての関連研究として、ピアノやオーケストラの楽譜に特化した研究では、飯島らが提案したオーケストラ譜からデュオ譜へ半自動編曲を行うシステム [1] や、藤田らが提案した、オーケストラの複数のパートで構成される楽譜に対して旋律とベースを推定し、楽譜を集約した後に演奏者の習熟度に応じたピアノ譜として生成するシステム [2] がある。また、GAを用いた研究として梶原らが提案した、現代音楽における作曲技法の一つ「12音技法」という技法を用いて、GAによってメロディを自動生成させる研究 [3] や、沼尾らが提案した、ユーザの設定値をもと

^{†1} 現在、日本大学

a) tahara@kthrlab.jp

b) uemura@kthrlab.jp

c) kitahara@chs.nihon-u.ac.jp

に GA を行い、即興の曲を生成する AMUSE というシステム [4] がある。GA を行うところは本研究と同じであるがこれらは評価関数に用いる制約が本研究と異なる点である。次に関連システムでは [5] のように最大同時発音数が 4 音という制約を破らずにファミコン風の曲を作ることができるシステムが存在したり、ファミコン風の音源を作るソフトシンセである [6] や [7] のようにファミコン風音楽に使用する音が既に用意されているものもある。しかしこれらは自動編曲するシステムではない。

これらに対して、本研究では既存の曲の重要な部分を抽出し、音楽未経験者でも気軽に既存の自分好みの曲をファミコン風音楽へ自動編曲することのできるシステムを提案する。ファミコン風音楽へ自動編曲するシステムを実現する上での課題は、原曲らしさを守りつつ、いかにファミコン音楽の制約を満たすように音を減らすかである。本研究では、原曲らしさやファミコン風音楽を満たすべき性質を定量的に定義し、それが最大になる音の省略方法を GA で探索する。

2. システムの概要

本研究のシステムは、既存のポピュラー曲の MIDI ファイルを入力とし、その MIDI ファイルを元にファミコン風音楽用に音削除を行った MIDI を生成するものである。音変換については、操作手順を見ながら用意された音を手動でユーザに割り当ててもらおう。ユーザに操作してもらうことにより、音量調節や音高調節などの自由度の高い操作が可能となる。また、入力とする曲は POP 曲の MIDI ファイルの中でもメロディが 1 パート目にある曲を対象とする。次にファミコン音楽の基本的構造を述べる。基本的にファミコン音楽の最大同時発音数は 4 音である。どのような音が使われているかを以下に示す。

矩形波

矩形波は計 2 音である。メロディやサブメロディに使われる。

三角波

三角波は 1 音である。三角波は 1 音。効果音の中でもキックやタムの三角波で作られることが多く、またベースラインにも使われる。

ノイズ波

ノイズ波は 1 音である。矩形波や三角波では作ることのできないその他の効果音はノイズ波で作られる。

これらの音を使い分け、ADSR の値を変えながら多種多様な音を作っている。ADSR とは、シンセサイザーの制御信号を設定するもので、音の始まりの音量値である Attack、アタック音から持続音の音量の変化である Decay、持続音を Sustain、持続音から音が消えるまでの時間 Release の 4 つの総称をいう。

ファミコン風音楽を作るにあたって重要なのは最大同時

発音数が 4 音という制約を守り、原曲らしさを損わせないことにある。本研究では、これらに着目をして研究を進めていく。

まず、原曲らしさを損わせないアレンジとは、以下のものである。

- アレンジ前の原曲とアレンジした曲との類似度が高いもの
- 単調でない音が残せていること
- メロディやリズムの基本的な構成が残せていること

アレンジする際にこの 3 つを守り、原曲から自動変換を行っていく。

ファミコン音楽の最大同時発音数は 4 音であり、特に和音が使われているパートは、1 パートだけでも最大同時発音数が 2 音や 3 音以上になってしまうことから音を極力削除する必要がある。また、原曲から自動変換を行っていく際に、全ての曲に対して同じ削除方法を適用するだけでは、元の楽曲のアレンジは様々であるため、どのパートを削除してどのパートを残すのかに関して、最適なルールをあらかじめ決めておくことは難しい。

そこで本研究では、様々な楽曲に対して適切な音の簡略化を行うため、簡略化を行った後のアレンジを満たすべき条件を定式化し、その式が最大になる簡略化を遺伝的アルゴリズム (genetic algorithm : GA) を用いて最も相応しいであろう組み合わせを決定する。GA とは、あるデータ表現を生成し、そのデータを交叉、突然変異などをランダムに組み替えをしながら、あらかじめ定義された適合度 (fitness 関数) の値が最大になるものを探すアルゴリズムである。

具体的には、まず、各パートに対してそれぞれ適切な音削除が行えるよう、各パートに対して音を全て単音に簡略化する「変換ルール」を設計する。各パートに適用することのできる変換ルールを、「ドラムパート以外に適用することができる変換変換ルール」が 7 つと「ドラムパート適用することができる変換ルール」が 5 つの計 12 個の変換ルールを設計する。メロディや伴奏以外にも、ドラムパートも適度に音数を減らさなければ、曲のリズムが正しく打たれていなかったり、ある打楽器音が集中してしまう可能性があるため、このようにドラムパートとそうでないパートで変換ルールを分ける手法をとる。

次に、最適化すべきパラメータを定義する。ここでは、各パートに適用する変換ルールと各パートの優先順位の組を GA の個体とし、変換後の音域やパート数に変換前にどの程度近いかなど、いくつかの観点から適合度関数を設計する。これらの定式化の下、遺伝的アルゴリズムを適用することで、各パートに適用すべき変換ルールと各パートの優先順位が決まるので、変換ルールに従って音を削減した後、優先順位が高いものから決められた音数まで音を残すことで、指定された制約を満たす原曲により近いアレンジを実現することができる。

3. 変換ルール

本システムは、任意の曲を入力とさせたときに、パートごとに対する変換ルールを適用させることにする。1つのパートにつき1つの変換ルールを適用することができる。各パートに適用することのできる変換ルールを、「ドラムパート以外に適用することができる変換ルール」と「ドラムパートに適用することができる変換ルール」の2つに分けて設定する。変換ルールの種類は、ドラムパート以外に使用することができる変換ルール(表1)が7個とドラムパートに使用することができる変換ルール(表2)が5つの合計12個とする。

また、ドラムパートには様々な音が含まれているため、各音にパート分解を行い、各音ごとに変換ルールを適用するものとする。

4. GAを行った曲の自動生成

4.1 ベースパートの推定

三角波に使用するベースパートを原曲のパートの中から決定する。最もノートナンバーの平均が低いパートがベースに選ばれるとの予想により、原曲からベースに選ぶパートは原曲の中でも最もノートナンバーの平均が低いパートをベースパートとすることにする。

4.2 個体の生成

個体は、パート番号と、そのパートに対する変換ルールをパートの個数組み合わせさせた1次元の配列であり、配列の添字の小さいパートが優先順位が高いものとする。パート番号を $a_i (1 \leq i \leq n)$ で表現したとき、GAの個体 $Chrom(G)$ を以下のような配列とする。

$Chrom(G) = \{ \text{パート数 } a_1, \text{パート } a_1 \text{ に対する変換ルール}, \text{パート数 } a_2, \text{パート数 } a_2 \text{ に対する変換ルール}, \dots, \text{パート数 } a_n, \text{パート数 } a_n \text{ に対する変換ルール} \}$
(配列の長さ = $a_n \times 2$)

三角波とノイズの同時発音は1音であるため、各波形内の音で優先順位の高いパートと音が被ってしまったらその音は削除するようにする。初期個体生成の段階では以下のような相応しくない個体が生成されないようにする。

- パート番号が入力とした曲のパート数を超過している
- 変換ルール番号が13個以上である
- パート番号が2つ存在する
- 同じパート番号が存在する

4.3 適合度関数

原曲を変換前、アレンジ後を変換後とした時のGAの適合度関数を決定していく。生成されたパートが次の条件を満たすとき、高い適合度を返すように設計する。

4.3.1 適合度関数 1 音高の平均値が変わらない

ある程度原曲と音高の平均値が近ければ、原曲により近い曲にアレンジがされるのではないかと考えるに基づいたものである。音高の平均値とは適合式を以下のように設計する。

$$\text{range}(G) = \frac{1}{\exp|(\text{変換前の音高の平均値}) - (\text{変換後の音高の平均値})|}$$

4.3.2 適合度関数 2 ドラム音以外の音符数が減っていない

原曲と同じ音符数を保っていた方が原曲により近いアレンジがされるのではないかと考えるに基づいて設計した適合度関数である。適合式を以下のように設計する。

$$\text{notesum}(G) = \frac{(\text{変換後の音符数})}{(\text{変換前の音符数})}$$

4.3.3 適合度関数 3 登場する楽器の種類が削られていない

原曲に登場するドラム音がなるべく削られていないものが同じリズムだという印象を与えるという考えに基づいて設計した適合式である。適合式を以下のように設計する。

$$\text{app}(G) = \frac{(\text{変換後の登場楽器数})}{(\text{変換前の登場楽器数})}$$

4.3.4 適合度関数 4 ドラム音以外の音が単調でない

ドラムの音が単調でない方が原曲の雰囲気を残せるのではないかと考えるに基づいて設計した適合度関数である。パート内に存在するすべてのノートの長さの集合を $\Omega = \{ \text{全音符}, \text{二分音符}, \text{四分音符}, \dots \}$ とし、 $E \in \Omega$ が出現する確率を $P(E)$ としたとき、適合式を以下のように設計する

$$\text{mon}(G) = \sum_{E \in \Omega} P(E) \log P(E).$$

4.3.5 適合度関数 5 メロディが残せている

曲の中で最も重要であるメロディパートを残すべきという考えに基づいた設計した適合度関数である。適合式を以下のように設計する。

$$\text{melody}(G) = \begin{cases} 0 & (\text{メロディが矩形波の中で最も優先順位が高い時}) \\ -\infty & (\text{そうでない時}) \end{cases}$$

4.3.6 適合度関数 6 ベースが残せている

ベースパートは三角波の中でも優先的に残すべきだという考えに基づいて設計した適合度関数である。適合式を以下のように設計する。

$$\text{bass}(G) = \begin{cases} 0 & (\text{ベースが三角波の中で最も優先順位が高い時}) \\ -\infty & (\text{そうでない時}) \end{cases}$$

4.3.7 適合度関数 7 一定のリズムパターンである

一定のリズムが乱れると全体の曲のまとまりに崩れが生じてしまうのではないかと考えるに基づいて設計した適

表 1 ドラムパート以外に適用することができる変換変換ルール

| 変換ルール番号 | 変換ルール名 | 変換内容 |
|---------|---------------------------------|--|
| 1 | 上の音だけ残す | 和音があった際に、和音の最も高い音を残す。 |
| 2 | 下の音だけ残す | 和音があった際に、和音の最も低い音を残す。 |
| 3 | 2 和音の 8 分音符分散和音 | 小節内の全ての音が 2 和音かつ同じ音だった際に、8 分音符の分散和音にする。 |
| 4 | 3 和音の 8 分音符分散和音 | 小節内の全ての音が 3 和音かつ同じ音だった際に、8 分音符の分散和音にする。 |
| 5 | 4 和音の 8 分音符分散和音 | 小節内の全ての音が 4 和音かつ同じ音だった際に、8 分音符の分散和音にする。 |
| 6 | 変換ルール 2 を適用した 4 分音符 全音符, 8 分音符化 | 和音があった際に、和音の最も高い音を残し、4 分音符以上の長さの音があったら 8 分音符に変換をさせる。 |
| 7 | 変換ルール 2 を適用した 4 分音符 全音符, 8 分音符化 | 和音があった際に、和音の最も低い音を残し、4 分音符以上の長さの音があったら 8 分音符に変換をさせる。 |

表 2 ドラムパートに適用することができる変換ルール

| 変換ルール番号 | 変換ルール名 | 変換内容 |
|---------|-----------------|--------------------|
| 8 | 8 分音符裏拍, 休符置き換え | 8 分音符の裏拍を休符に置き換える。 |
| 9 | 8 分音符表拍, 休符置き換え | 8 分音符の表拍を休符に置き換える。 |
| 10 | 4 分音符裏拍, 休符置き換え | 4 分音符の裏拍を休符に置き換える。 |
| 11 | 4 分音符表拍, 休符置き換え | 4 分音符の表拍を休符に置き換える。 |
| 12 | 何もしない | そのまま音を残す。 |

合度関数である。

変換ルールを適用した後の、ドラムパート内の楽器 i の音符列に対して、隣り合う 2 つの IOI (inter-onset interval) の集合を $\Omega_2 = \{(1.0, 1.0), (0.5, 1.0), \dots\}$ とする。また、1.0 を四分音符 1 つ分の長さとする。このとき、もっとも頻繁に出現するものの出現確率: $\max_{E \in \Omega_2} P(E)$ が十分に高ければ、この曲は一定のリズムが用いられていると考えることができる。そこで、適合式を以下のように設計する。

$$\text{rhythm}(G) = \max_{E \in \Omega_2} P(E)$$

4.4 適合度関数の総適合度

設計した条件を満たすように、GA から得られる 7 個の適合度関数 $F(G)$ を次のように定義する。

$$F(G) = \text{range}(G) + \text{notesum}(G) + \text{app}(G) + \text{mon}(G) \\ + \text{melody}(G) + \text{bass}(G) + \text{rhythm}(G).$$

4.5 MIDI データ生成

GA で生成された最適解をもとにパートを生成する。出力される MIDI ファイルのパートは、1,2 パート目が矩形波となるパートで、変換ルールを適用した形で適合度の高かった矩形波 2 パートが順に入れられる。3 パート目には、出力された矩形波 1,2 パートで音が入っていない箇所が存在した時に矩形波の優先順位の 3 番目以降の音を順次追加していくパートとなる。

次に三角波の優先順位が高いものから順にパートを追加していく。三角波と同じように、ノイズも優先順位の高いものから順に追加される。

5. 評価実験

生成されたファミコン風音楽が原曲に近く、かつ音楽性を保っているかについての評価方法および結果と考察を述べる。どの手法が最も適切な編曲がされているか以下の 4 つのアレンジ手法を用意し、主観評価と客観評価に分けて評価を行った。

アレンジ 1 変換ルールありで小節ごとに GA を行う。

アレンジ 2 変換ルールありで楽曲全体で GA を行う。

アレンジ 3 メロディとベースだけを決定後、GA を行わず、ランダムに決定した変換ルールを各パートに適用させる。また、パートの優先順位はメロディとベースを最優先とし、その他をランダムとする。

アレンジ 4 変換ルールなしで小節ごと GA を行う。

評価対象の曲は、20 小節分ある J-POP の曲 15 曲を対象とした。しかし、主観評価で 15 曲中の 1 曲のベースパートがうまく生成されていないという問題が生じたため、主観評価では 1 曲を差し引いた 14 曲を対象とした。世代数は 20 と設定する。

5.1 主観評価

5.1.1 評価方法

本システムによって生成されたファミコン風音楽が適切に生成されていたか、音楽経験者 1 名に評価を行ってもらった。

原曲を 1 度聴いたあとに、これらの条件でアレンジされた曲を再生し、7 点満点の点で評価をしてもらった。原曲とアレンジされた曲は何度も聴いても良いものとし、質問に対するコメントがあれば記入してもらった。質問

項目は以下に示す。

質問 1 自然なリズムでしたか

質問 2 音のつながりに違和感がないと感じましたか

質問 3 音の主旋律、ベース以外のパートの選び方が適切でしたか

質問 4 原曲と近いと感じましたか

4つの質問に対して15曲の平均値を、表3-表6に示す。

5.1.2 結果、考察

質問1は全体的に平均点の低いアレンジ2では14曲中3曲で評価が3以下であったのに対し、平均点の高いアレンジ1では評価が3以下の曲が0曲、アレンジ4では1曲と、アレンジ2よりも比較的高い評価が得られた。アレンジ2の低い評価が得られた曲では「入ったり入らなかったりするパートが不自然」「細かいリズムが途切れ途切れになっている」とのコメントがあり、またアレンジ3の低い評価が得られた曲では「リズム担当が雑念としていて断片化され自然な流れがない」「細かいリズムが強調されすぎている」というコメントがあった。

質問2は全体的に平均点の低いアレンジ2では14曲中5曲で評価が3以下であったのに対し、平均点の高いアレンジ3では14曲中1曲で評価3以下、アレンジ4では3曲で評価が3以下という評価が得られた。アレンジ1の低い評価が得られた曲では「ピッチのつながりはいいがベースが聞こえない」とのベースの音域についてのコメントや、「全音音階の装飾が強調されすぎている」との指摘を受けた。また、アレンジ2の低い評価が得られた曲では「メロディックな声部が後半にいくに従って消えてリズム声部になってしまっている」「極端にリズムが変わる」というリズムの偏りと「冒頭部分が聞こえない」という音選びが適切でないため違和感を感じたと指摘を受けた。

質問3は全体的に平均点の低いアレンジ2では14曲中9曲で評価が3以下であったのに対し、平均点の高いアレンジ3では14曲中1曲で評価3以下、という評価が得られた。アレンジ2の低い評価が得られた曲では「リズム性部が強調されすぎていて必要な音が欠けている」「ベースはいいが対旋律が物足りない」音の重なり不足について指摘を受けた。アレンジ4の低い評価が得られた曲では「主旋律以外のパートが断片的であった」とのコメントがあった。

質問4は全体的に平均点の低いアレンジ2では14曲中9曲で評価が3以下であったのに対し、平均点の高いアレンジ3では14曲中4曲で評価3以下、アレンジ4では6曲で評価が3以下、アレンジ1では4曲で評価が3以下という評価が得られた。アレンジ2の低い評価が得られた曲では「途中の装飾声部の曲調が原曲とはかけ離れている」とのコメントが多かった。アレンジ3の低い評価が得られた曲では「高音の絡みが曲全体として偏りの印象を与える」とのコメントがあった。

全体的にアレンジ2の評価が低かったのは曲全体に対し

てGAを行い変換ルールを適用しているため、パート内に優先すべきもの、そうでないものが混ざっているからと考えられる。また、アレンジ1やアレンジ4がランダムで変換ルールを決定するアレンジ3より評価が低くなった。このような結果となった理由として装飾音に動きがあると主旋律が聞き辛くなっていることが原因として挙げられる。適合度関数4である“ドラム音以外の音が単調でない”音を選ぶ様に設計したものが裏目に出てしまったからだと考えられる。また、アレンジ3はメロディとベース以外はランダムに優先順位も変換ルールも決定されるため、偶然良いアレンジが生まれてしまったことも考えられる。

表 3 質問1の各アレンジの評価

| | アレンジ1 | アレンジ2 | アレンジ3 | アレンジ4 |
|-------|-------|-------|-------|-------|
| 1 曲目 | 6 | 7 | 6 | 6 |
| 2 曲目 | 5 | 3 | 5 | 6 |
| 3 曲目 | 7 | 6 | 6 | 6 |
| 4 曲目 | 6 | 6 | 6 | 6 |
| 5 曲目 | 5 | 6 | 6 | 6 |
| 6 曲目 | 6 | 6 | 6 | 6 |
| 7 曲目 | 6 | 6 | 6 | 6 |
| 8 曲目 | 6 | 6 | 7 | 6 |
| 9 曲目 | 5 | 7 | 3 | 7 |
| 10 曲目 | 5 | 3 | 6 | 6 |
| 11 曲目 | 6 | 6 | 6 | 6 |
| 12 曲目 | 5 | 2 | 6 | 3 |
| 13 曲目 | 6 | 6 | 6 | 7 |
| 14 曲目 | 7 | 6 | 3 | 6 |
| 平均 | 5.8 | 5.3 | 5.6 | 5.9 |

表 4 質問2の各アレンジの評価

| | アレンジ1 | アレンジ2 | アレンジ3 | アレンジ4 |
|-------|-------|-------|-------|-------|
| 1 曲目 | 5 | 2 | 7 | 5 |
| 2 曲目 | 6 | 6 | 5 | 5 |
| 3 曲目 | 6 | 5 | 5 | 3 |
| 4 曲目 | 3 | 6 | 6 | 7 |
| 5 曲目 | 5 | 6 | 3 | 7 |
| 6 曲目 | 2 | 2 | 5 | 4 |
| 7 曲目 | 5 | 6 | 6 | 3 |
| 8 曲目 | 6 | 5 | 5 | 7 |
| 9 曲目 | 5 | 7 | 6 | 6 |
| 10 曲目 | 5 | 3 | 6 | 6 |
| 11 曲目 | 7 | 2 | 6 | 5 |
| 12 曲目 | 5 | 3 | 6 | 3 |
| 13 曲目 | 6 | 6 | 7 | 7 |
| 14 曲目 | 6 | 3 | 3 | 5 |
| 平均 | 5.1 | 4.4 | 5.4 | 5.3 |

5.2 客観評価

5.2.1 評価方法

本システムによって生成されたファミコン風音楽が適合度関数の値を満たしているかの評価を行う。1曲につき全20小節、それぞれ曲全体の適合度を求める。以下の調査4つを行った。

調査 1 ドラム音以外の音符数が減っていないか。

調査 2 登場する楽器の種類が削られていないか。

表 5 質問 3 の各アレンジの評価

| | アレンジ 1 | アレンジ 2 | アレンジ 3 | アレンジ 4 |
|-------|--------|--------|--------|--------|
| 1 曲目 | 6 | 1 | 7 | 3 |
| 2 曲目 | 6 | 2 | 6 | 6 |
| 3 曲目 | 6 | 3 | 5 | 3 |
| 4 曲目 | 3 | 5 | 6 | 6 |
| 5 曲目 | 6 | 6 | 3 | 7 |
| 6 曲目 | 3 | 3 | 5 | 5 |
| 7 曲目 | 3 | 6 | 6 | 3 |
| 8 曲目 | 6 | 3 | 7 | 7 |
| 9 曲目 | 3 | 5 | 6 | 3 |
| 10 曲目 | 5 | 3 | 6 | 3 |
| 11 曲目 | 7 | 3 | 7 | 3 |
| 12 曲目 | 3 | 3 | 6 | 3 |
| 13 曲目 | 2 | 3 | 5 | 6 |
| 14 曲目 | 6 | 6 | 7 | 7 |
| 平均 | 4.6 | 3.7 | 5.9 | 4.6 |

表 6 質問 4 の各アレンジの評価

| | アレンジ 1 | アレンジ 2 | アレンジ 3 | アレンジ 4 |
|-------|--------|--------|--------|--------|
| 1 曲目 | 5 | 2 | 6 | 3 |
| 2 曲目 | 5 | 3 | 5 | 6 |
| 3 曲目 | 6 | 3 | 5 | 3 |
| 4 曲目 | 5 | 5 | 3 | 6 |
| 5 曲目 | 5 | 6 | 3 | 6 |
| 6 曲目 | 3 | 3 | 5 | 5 |
| 7 曲目 | 5 | 6 | 6 | 3 |
| 8 曲目 | 6 | 3 | 7 | 7 |
| 9 曲目 | 3 | 6 | 3 | 6 |
| 10 曲目 | 3 | 3 | 6 | 3 |
| 11 曲目 | 6 | 3 | 6 | 3 |
| 12 曲目 | 3 | 3 | 6 | 3 |
| 13 曲目 | 6 | 6 | 7 | 7 |
| 14 曲目 | 6 | 3 | 3 | 5 |
| 平均 | 4.8 | 3.9 | 5.0 | 4.7 |

調査 3 ドラム音以外の音が単調でないか。

調査 4 一定のリズムパターンであるか。

4つの調査に対して評価を行った1曲ごとの結果と15曲の平均値、生成された一部の楽譜を、図1-8に示す。次に、4つの調査について4つのアレンジを各群としたときに群全体に対してクラスカル・ウォリス検定を行った。有意水準を0.05とした時、調査1のP値は0.18、調査3のP値は0.19、調査4のP値は0.99、だったため、この3つは有意差は見られず、調査2のP値は0.000033であったため、調査2では有意差が見られる結果となった。また、有意水準を0.05とした時、群全体で有意差があった調査2に対して、Bonferroni法を使い、群同士の有意差を求めた。2群の組み合わせが6通り存在するため、P値が0.0083以下であれば有意差ありとみなす。結果はアレンジ1とアレンジ2、アレンジ2とアレンジ4で有意差がみられる結果となった。

5.2.2 結果、考察

調査1では、4.3の適合度関数2に基づいた評価を行った。図1の15曲の平均値をみると、4つのアレンジ手法でドラム音以外の音符数の変化に大きな違いはない結果となった。矩形波2音は、優先順位の高い2つのパートだけを抽出しているわけではなく、図2のように、優先順位の高い2つのパートの中に休符が存在した際、3番目以降に優先順

位の高いパートの音を順に間に追加していく手法をとっている。この手法により、優先順位の高い2パートの音の鳴っていない箇所をできるだけ無くすようにしている。そのため、大きな違いはなかったのではないかという予想がつく。

調査2では、4.3の適合度関数3に基づいた評価を行った。この調査では、小節ごとに登場する楽器の種類は減っていないかの調査をする。図3の15曲の平均値を見ると、アレンジ2の平均値が最も低いことから、アレンジ2が登場楽器が最も減っていることがわかる。アレンジ2はパートごとにGAを行うため、特定のドラムが1つのパートを通して1音登場するだけでも、適合度が上がる様になっている。このことにより、小節ごとにGAを行うよりも1小節間に登場する楽器の種類が削られていると考えられる。例えば、例4のように、小節ごとにGAを行ったアレンジ1とパートごとにGAを行ったアレンジ2の楽譜の一部を比較してみると、アレンジ1の楽譜の方が小節ごとに見ると登場楽器数が多いことがわかる。次に、図3から、15曲の平均値がアレンジが低い結果となった。これは、適合度を考慮しないGAを行わなかったためだと思われる。

調査3では、4.3の適合度関数4に基づいた評価を行った。図5の15曲の平均値から、小節ごとにGAを行うアレンジ1, 3, 4に比べ、GAを行うアレンジ2が最も単調な音が選ばれやすいという結果になった。最も適合度の値の差が大きかったアレンジが使われている曲の一部を図6に示す。2番目のパートに注目すると、パートごとにGAを行うアレンジ2では全音符が2小節分使われていて、小節ごとにGAを行ったアレンジ4の方は全音符が選ばれることなく複雑な音が選ばれていることがわかる。また、これはアレンジ1とアレンジ3もアレンジ4と同様に小節ごとに処理を行っているため全音符以上の長さの音が選ばれにくい。それに対し、アレンジ2は、全パートを通して複雑であるかを比較しているため、このように2小節全音符が続くといった様な非常に単調な音の並びが含まれている可能性がある。客観評価でもアレンジ2に対して「メロディックではない声部が選ばれている」「イントロのメロディーラインが適切でない」と、単純な音並びが混ざっていることが原因で音選びが適切でないことを指摘するコメントもあった。そのため、図5の15曲の平均を見てもアレンジ2は最も複雑な音が選ばれていない結果となった。

調査4では、4.3の適合度関数7に基づいた評価を行った。アレンジ2の値が高い結果となったのは、小節ごとに違った変換ルールを適用させるのではなく、曲全体に対して同じ変換ルールを適用させるため、小節ごとに別の変換ルールを適用するパートよりも一定のリズムが刻まれているためだと思われる。例として、最も適合度の値の差が大きかったアレンジが使われている曲の一部と各アレンジのハイハットで適用された変換ルールを図8に示す。小節ごとに変換ルールを決定するアレンジのハイハットパートに

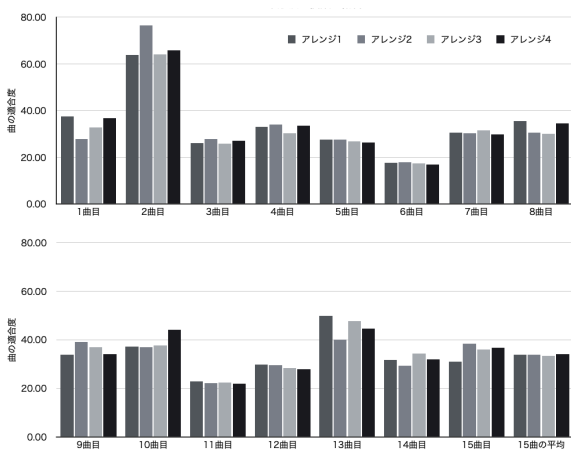


図 1 調査 1. ドラム音以外の音符数が減っていないか.

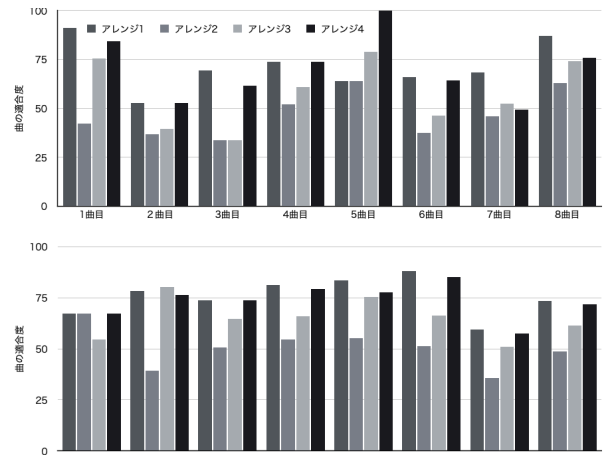


図 3 調査 2. 登場する楽器の種類が削られていないか.



図 2 音追加イメージ (7 曲目の一部)

注目すると、小節ごとに GA を行うアレンジ 1 では 1 小節目と 4 小節目に変換ルール 13 の「そのまま音を残す」が選ばれ、2 小節目と 3 小節目には変換ルール 9 である「8 分音符表拍、休符置き換え」が選ばれている。それに対し、アレンジ 2 では、1 パートに同じ変換ルールを決定するため、ハイハットの全小節に対し変換ルール 8 である「8 分音符裏拍、休符置き換え」が選ばれている。このように、小節ごとと異なった変換ルールを適用するアレンジ 1 よりも、パートごとに同じ変換ルールを適用されるアレンジ 2 の方がより長い時間一定のリズムを刻まれていることになる。アレンジ 2 の客観評価のコメントでも「一貫してシンプルである」「基本リズムは一定していて自然である」とリズムの一定さに対して同様なことが言われている。

適合度関数の値を満たしているかの評価では、GA を行った 3 つの手法よりも GA を行わなかったアレンジ 3 の評価が最も総合的に低いことが確認できた。またその中でも、曲全体に対して GA を行うアレンジ 2 よりも、小節ごとに GA を行うアレンジ 1 とアレンジ 4 を用いたアレンジが総合的に高い評価が得られた。

6. 結論

本研究では、既存の J-POP 曲をファミコン風音楽へアレンジするシステムを提案した。最大同時発音数が 4 音という制約を守るため、原曲からどのように原曲らしさを損わない音を選択するか決めなければならない問題点があった。そこで、音を減らすために各パートに対する変換ルールを

原曲

アレンジ 1: 変換ルールありで小節ごとGAを行う。

アレンジ 2: 変換ルールありで楽曲全体でGAを行う。

図 4 調査 2. アレンジ 1 とアレンジ 2 の登場楽器の比較 (1 曲目の一部)

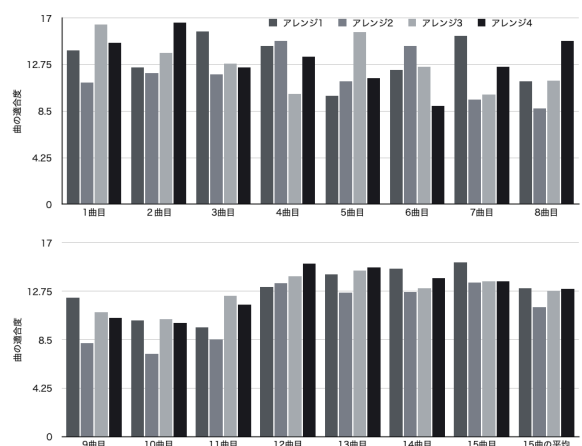


図 5 調査 3. ドラム音以外の音が単調でないか.

12 個用意した。全ての曲に決められた変換ルールを適用させるだけでは音楽も単純なものになってしまう問題点を、GA を行い、パートごとに適用させるルールを機械的に決

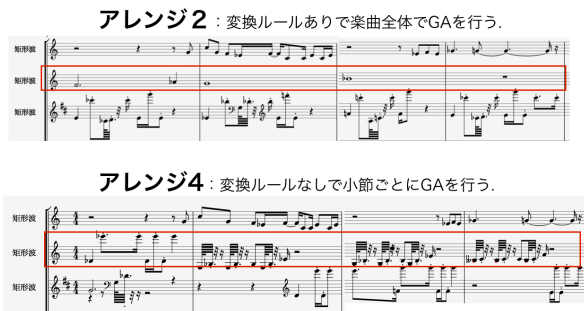


図 6 調査 3. アレンジ 2 とアレンジ 4 の矩形波の音の比較 (8 曲目の一部)

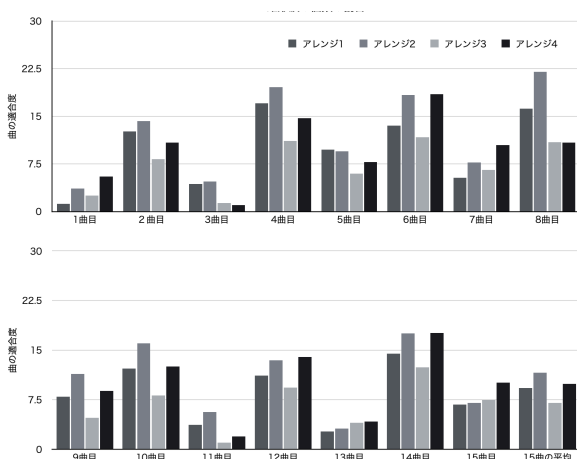


図 7 調査 4. 一定のリズムパターンであるか。



図 8 調査 4. アレンジ 1 とアレンジ 2 のドラム音リズム比較 (8 曲目の一部)

定することによって解決した。

主観評価では、GA を行わず、ランダムに決定した優先順位と変換ルールを各パートに適用させるアレンジ 3 が最も高い評価が得られた。ランダムに優先順位と変換ルールを

決定するアレンジ 3 に対し GA を行うアレンジ 1 とアレンジ 4 の評価が低かった原因として、装飾音に動きがあると主旋律が聞き辛くなってしまおうという意見があったことから、“ドラム音以外の音が単調でない”音を残す適合度関数 4 が裏目に出てしまった部分であった。客観評価では、特に変換ルールを用い、小節ごとに GA を行うアレンジ 1、続いて変換ルールを用いず、小節ごとに GA を行うアレンジ 4 の手法は最も精度の高い結果となった。この 2 つの評価実験では特にこの手法が良かったという結果は総合的には得られなかった。

6.1 今後の展望

システムにおいて、音変換については、操作手順を見ながら用意された音を手動でユーザに割り当ててもらおうシステムとなったが、音を制御してもらうのは自由度は高い一方、音変換の際に手間かかってしまうため、音変換まで自動でできる様にシステムを改造させる。

また、本研究では音の削除を主に行ってきたが、今後は最大同時発音数が 4 音未満の楽譜から音を追加できるようなアレンジが生成できるようにする。

謝辞 本研究は、JSPS 科研費 19K12288, 16H01744 から支援を受けた。

参考文献

- [1] 飯島 崇裕, 齋藤 康之: “オーケストラ譜からデュオ譜への半自動編曲手法に関する研究”, 映像情報メディア学会技術報告, pp19-22, 2013.
- [2] 藤田 顕次, 大野 博之, 稲積宏誠: “習熟度を考慮した複数楽譜からのピアノ譜生成手法の提案”, 情報処理学会研究報告, 音楽情報科学, 2008-MUS-077, pp47 - 52, 2008.
- [3] 梶原 悠介: “遺伝的アルゴリズムを用いた 12 音技法に基づく自動作曲手法に関する研究” ファジィシステムシンポジウム, p792-803, 2009.
- [4] 沼尾正行, 高木将一, 中村啓佑: “ユーザの感性に合わせた自動編曲及び作曲”, 情報処理学会研究報告, 音楽情報科学, 2001-MUS-041, pp49-54, 2001.
- [5] BleuBleu: “FamiStudio”, <https://github.com/BleuBleu/FamiStudio>
- [6] Yokemura: “Magical8bit Plug”, <https://ymck.net/>
- [7] mu-station: “mu-station”, <http://mu-station.chillout.jp/>
- [8] “NES-MDB dataset”, <https://github.com/chrisdonahue/nesmdb>