

## 遺伝的アルゴリズムを用いたメロディー進行と リズムの組合わせによる自動作曲

田中 健<sup>†</sup> 外山 史<sup>†</sup> 東海林 健二<sup>†</sup>

<sup>†</sup>宇都宮大学工学部

<sup>†</sup>〒 321-8585 宇都宮市陽東 7-1-2

E-mail: tanaka@athena.is.utsunomiya-u.ac.jp

あらまし 本論文では、作曲を、データベースにある曲のリズムと音高パターンの最適な組合わせを見つける最適化問題とみなし、これを遺伝的アルゴリズム (GA) を用いて解く手法を提案する。データベースには、起承転結のような構造の分かりやすい曲 (童謡、唱歌など) を格納した。そして、別々の曲のリズムと音高パターンを小節ごとに組み合わせてできた曲を GA の評価対象とした。提案した手法により曲を生成した実験結果を示す。

キーワード 自動作曲, 遺伝的アルゴリズム

## Automatic Composition by Combination of Pitch Transition Patterns and Rhythm Using a Genetic Algorithm

Takeshi Tanaka,<sup>†</sup> Fubito Toyama,<sup>†</sup> and Kenji Shoji,<sup>†</sup>

<sup>†</sup>Faculty of Engineering, Utsunomiya University

<sup>†</sup>7-1-2 Yoto, Utsunomiya 321-8585, Japan

E-mail: tanaka@athena.is.utsunomiya-u.ac.jp

**Abstract** In this paper, we regard an automatic composition as an optimization problem which finds an optimum combination of rhythm and pitch transition patterns in the database. We selected the pieces of music that are simple in structure from the children's songs, the ministry of education songs, etc., and we stored them into the database. The fitness function of a GA evaluates the composed pieces of music that are sequences of the combination of rhythm patterns and pitch transition ones taken from the database in each measure. The pieces of music composed by the proposed method are shown.

**keywords** automatic composition, genetic algorithm

# 1 まえがき

音楽として人間が認めることのできる自然な曲を計算機により自動的に作曲をするという試みは古くから行われてきた [1] ~ [5]。

このような自動作曲を行う手法として、まず推移確率と情報理論による手法があげられる [4][5]。これは確率論的な音楽分析を行うことにより作曲している。しかし、これらの手法では情動を喚起する仕組みが分析、プログラミングされていないため、おもしろさ、興奮を引き起こすような作品ができにくいという問題がある。

一方、遺伝的アルゴリズム (Genetic Algorithm; GA) を用いて作曲を行う手法 [1][2] がある。GA は交叉・突然変異の遺伝操作と自然淘汰の機構をモデル化した探索手法であり、最適化問題においてその評価関数に基づいて解の候補の集団が進化していく過程に解の探索過程を置き換えた最適化アルゴリズムである。GA は、常に複数の解を保有しながら探索を行うので局所的な解に陥りにくく、大域的な探索能力に優れている。作曲を、膨大な数の音符の組み合わせの中から、最適な組み合わせを探索する最適化問題と考えると、GA は作曲に適した最適化アルゴリズムであると考えられる。

今井らの手法 [1] は、実際の作曲家が良く行うように、まず、コード進行を決定し、その決定されたコード進行に基づきメロディーを生成している。音楽理論に基づいた評価値 (適応度) を用いてコード進行とそれぞれのコードを構成する音を GA により最適化している。同様にメロディーを生成するプロセスにおいても、GA により対応するコードを参照しながら最適化を行っている。この手法では、音楽理論的には正しい曲を生成することができるが、自然で面白味のある曲が生成できるとは言いがたい。

本論文では、多くの人にとって構造が分りやすく自然な曲を作曲することを目的とし、GA と楽曲によるデータベースを用いた、リズムと音高パターンの組み合わせによる自動作曲の手法を提案する。具体的には、構造のはっきりした面白味のある曲を生成することを目的として曲に「起承転結」の構造を用いる。構造の分りやすい数多くの楽曲 (童謡、唱歌など) のデータを 1 曲ごとに「起承転結」の 4 つの部分に分け、それぞれを分類して格納したデータベースを作成し、1 小節ごとにさまざまな曲のリズム、音高パターンを別々に「起承転結」ごとにデータベースから取り出して組み合わせ、連結させ



図 1: 本手法で用いる曲の構造

ることにより、新しい曲を生成する。この組み合わせを GA を用いて最適化することにより自動作曲を行う手法を提案する。

## 2 曲の構造と表現

本手法で提案する自動作曲の方法の概念について説明する。本手法では、作曲を行う際、多数の楽曲のデータを格納したデータベースを作成し、使用する。データベースには、曲の音符列のデータを 1 小節分ごとに格納し、作曲を行う際にはこのデータベース中から 1 小節分ごとにリズムの情報と音高パターンの情報をそれぞれ別々に取出し、それを合成することによって 1 小節分の音符列を作成する。それを  $m$  小節分繰り返し ( $m$  は生成する曲の小節数) つなげることによって新しい曲を生成する。この組み合わせを GA を用いて最適化することによって自動作曲を行う。

本研究では、 $m$  小節からなる 4 拍子の、構造の分りやすい曲を生成する。人間が聞いて面白さを感じることができる曲を生成するという目的から、本手法では曲に、「起承転結」の形式の構造を用いる。つまり、図 1 のように、1 曲を「起」「承」「転」「結」の 4 つの部分の連結で構成し、「起」には曲の導入部としての、「承」には「起」の要素の受け継ぎとしての、「転」には曲の展開部としての、「結」には曲の完結部としての役割をそれぞれ持たせる。楽曲には、この「起承転結」の構造を持つものが多数存在する。よって、データベースに格納する曲はこの構造を持つ曲を使用し、データベース作成時にもこの構造を反映させる。

### 2.1 データベース

本研究では、データベースを作成する際、曲の

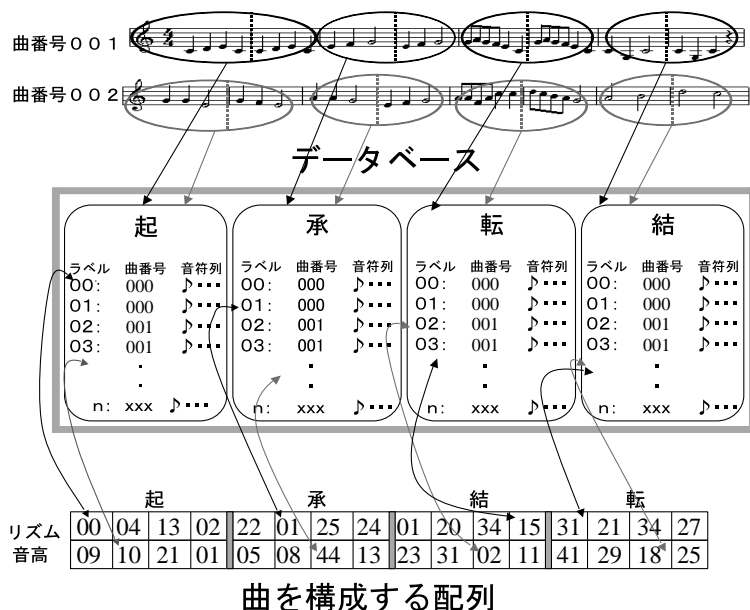


図 2: 元の曲とデータベースと生成される曲を表す配列との関係

データを全て対話的に「起」「承」「転」「結」に分類し、データベースに格納する。具体的には、格納する曲のデータに1曲ごとに曲番号をつけ、曲を「起」「承」「転」「結」の部分ごとに分類した後、八長調に移調し、この音符列の情報をさらに1小節ごとに分断して、「起」「承」「転」「結」ごとに1小節分ずつラベル番号をつけてデータベースに格納する(図2参照)。つまり、データベース内の「起」「承」「転」「結」ごとに格納されている1小節ごとの音符列の情報には、図2に示すように、ラベル番号と、元の曲による曲番号がつけられていることになる。

## 2.2 データベースを用いた曲の表現方法

本研究では、データベース内のさまざまな曲のリズムと音高パターンを「起」「承」「転」「結」ごとに1小節分ずつ組み合わせることによって作曲を行う。

本研究で生成される曲の1曲分の情報はリズムのラベル番号と音高パターンのラベル番号それぞれ  $m$  小節分から成る、 $2 \times m$  の整数型の2次元配列で表す。したがって、1小節ごとに配列内のラベル番号に対応する音符列のリズムと音高パターンをデータベースからそれぞれ別々に取り出し、これらを合成することにより新しい音符列を生成することができ、これを  $m$  小節分繰り返してつなげることによ

て新しい曲を生成することができる(図2参照)。なお、音高パターンの情報には元の曲のメロディーのコード(和音)の情報も含まれ、これを作曲時に使用する。例えば、図2で、曲を構成する配列の第5小節目(「承」の第1小節目)は、リズムにおいてはデータベース中の「承」の部分のラベル番号22番の音符列のリズムを、音高パターンにおいては、ラベル番号05の音符列の音高パターンをそれぞれ取り出し、これらを合成して新しい曲の第5小節目を生成する。

## 2.3 リズムと音高パターンの合成処理

本手法では、リズムと音高パターンを合成する際には、それぞれの要素を先頭から順に1つずつ対応させ、これをリズムの要素が無くなり、4分音符4つ分の長さの音符列が生成されるまで続けることによって合成を行う。この過程において、4分音符4つ分の長さの音符列が生成され、音高パターンの要素がリズムの要素に比べ余った場合は、余った音高パターンの要素を全て無視して合成処理を終了し、次の小節の合成処理に新たに移る(図3参照)。また、音高パターンの要素が、リズムの要素に比べて足りない場合、つまりリズムの要素が余ってしまう場合には、合成の終了した音高パターンの要素を始めから繰り返して、残ったリズムの要素の余った

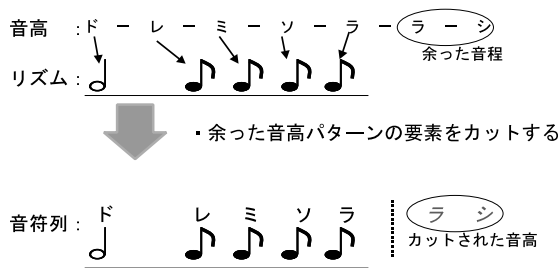


図 3: 音高パターンの要素が余った場合

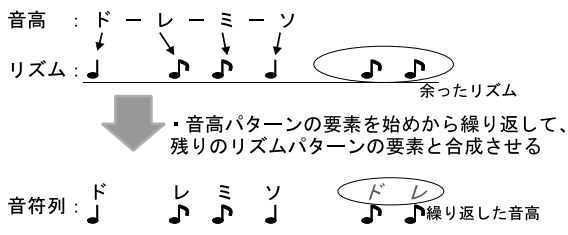


図 4: 音高パターンの要素が足りない場合

部分と順に1つずつ合成することによって処理を続行し、リズムの要素すべてに音程が対応、4分音符4つ分の長さの音符列が生成された時点で合成処理を終了し、次の小節の合成処理に新たに移る(図4参照)。

### 3 GAの自動作曲への適用

#### 3.1 染色体の定義

染色体を1つの曲に対応させる。1つの曲は2.2で述べたように、データベースを参照するためのラベル番号が格納された配列で表される。曲はリズム用の配列と、音高パターン用の配列という2つの配列によって表されるため、染色体は $2 \times m$ の2次元整数型配列で表される(図5参照)。

#### 3.2 初期染色体の発生

「起」「承」「転」「結」それぞれ1小節ごとにリズム用と音高パターン用のラベル番号を全てランダムに発生させ配列に格納し、それを初期染色体とする。

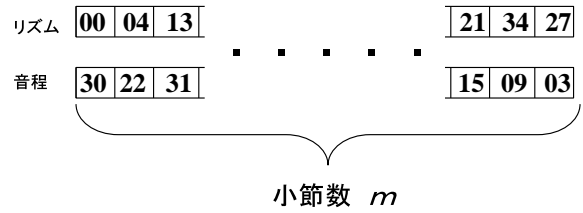


図 5: 染色体となる配列

#### 3.3 適応度の計算

本研究では、曲は全て八長調として扱うこととする。本研究では楽曲データを格納したデータベースを作曲に使用しているため、文献[1][2]で評価基準としていた、曲中のメロディーがコードに正しく対応しているか、メロディーがある程度なめらかであるか、という問題に関しては解消されている。

本研究では、以下に定めた評価方法に基づいて個体に項目ごとの得点を与える。そして、各項目ごとの得点に重み付けをして統合したものを個体の適応度とする。本研究での評価基準を以下に示す。

##### < 1 > コード進行について

- 「起」の最初のコードと「結」の最後のコードが「C」または「Am」である個体を高く評価する。曲の始まりと終わりを形作るためである。「起」の最初のコードの得点を *Firstchord*、「結」の最後のコードの得点を *Endchord* とし、条件を満たした場合はそれぞれ加算する。
- 「転」の部分のコード進行について、曲の展開部としての役割を持たせるため、「起」「承」の部分とは異なる進行となる個体を高く評価する。この項目の得点を *Turnchord* とする。
- コード進行にはお互いに相性があり、相性の良いコード進行で曲を構成するのが良い。この項目の得点を *Badchord* とし、相性の良くないコード進行があった場合、減点していく。

##### < 2 > 旋律の類似性について

- 曲の「起承転結」の構造を明らかにしつつ、曲全体の統一感を得るために、旋律の類似性に着

目した評価を行う。まず、「起」の旋律と「承」の旋律が類似している個体を高く評価する。また、「転」の旋律が「起」「承」の旋律とある程度類似していない個体を高く評価する。さらに、「結」の旋律が「転」の旋律とある程度類似していない個体を高く評価する。この項目の得点を *Similarity* とし、それぞれの場合においてある一定の類似度を定め、その値に近いほど高い得点を与える。ここで旋律の類似度は、DP マッチング法によりリズム、それぞれの旋律中での 2 音間の音高の差分の推移、2 旋律間の音高の差の 3 つの項目について判定し、統合した値を使用した。

### < 3 > 終りの音の長さについての評価

- 曲の終止感を得るため、終わりの音の長さが 2 分音符以上である個体を高く評価する。この項目の得点を *Endnote* とし、条件を満たす場合加点する。

### < 3 > 曲番号について

- 同じ小節内で、同じ曲番号のリズムと音高パターンの組み合わせとなる個体を大幅に低く評価する。これはデータベースに格納した曲と同じ音符列が生成されることを防ぐためである。この項目の得点を *Number* とし、条件を満たす場合、減点する。

これら全てを重み付けをして統合した関数  $F$  が本研究における適応度関数である。

$$\begin{aligned}
 F = & W_1 \cdot Firstchord + W_2 \cdot Endchord \\
 & + W_3 \cdot Turnchord + W_4 \cdot Badchord \\
 & + W_5 \cdot Similarity + W_6 \cdot Endnote \\
 & + W_7 \cdot Number \quad (1)
 \end{aligned}$$

## 3.4 選択交配

交叉する個体の選択には、適応度に比例した確率で選択を行うルーレット選択法を採用する。また、個体の集団中で、適応度の高い上位数個体をそのまま次世代に残すエリート保存戦略を採用する。この方法には、各世代で最もよい適応度を有する解を常に継承することができ、交叉や突然変異によって破壊されない、また、解の収束を速めるという利点がある。

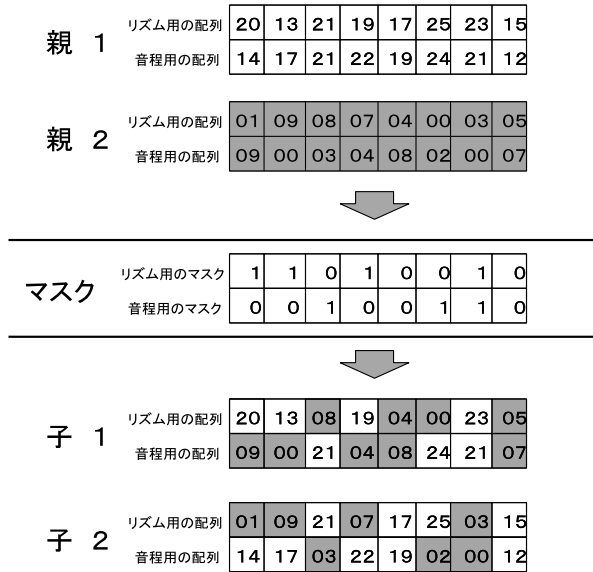


図 6: 本手法での一様交叉の例

## 3.5 交叉

交叉は、2 つの親の染色体を組み替えて子の染色体をつくる操作である。本手法での交叉の方法は、図 6 のような、交叉時にマスクをかけてそれによってどちらの親の遺伝子を受け継ぐかを決定する一様交叉である。図 6 に示すように、2 つの親となる個体及びマスクを設定する。マスクのビットが 0 の時には子 1 には親 1 の遺伝子をコピーし、マスクのビットが 1 の時には親 2 の遺伝子をコピーする。子 2 に関しては、これの逆を行う。これを選択された親の、リズムパターン用の配列と音程のパターン用の配列に対して別々に行う。

## 3.6 突然変異

突然変異は、染色体の一部をある確率で変化させる操作である。突然変異を起こすことによって、交叉だけでは得られないパターンを生成することができる。また、突然変異は設定した突然変異率に従って個体ごとに起こるものとし、選ばれた個体中のリズムと音高パターンでそれぞれ別々にランダムで選ばれるあるラベル番号を、ランダムに他のラベル番号に置き換えるものとする。



図 7: 自動作曲された 8 小節の曲



図 8: 自動作曲された 16 小節の曲

## 4 実験

対話的に作成した楽曲 41 曲によるデータベースを用意し、これを用いて実験を行った。本実験では、起承転結がそれぞれ 2 小節ずつの 8 小節の曲と、起承転結がそれぞれ 4 小節ずつの 16 小節の曲を作成した。実験時のパラメータはそれぞれ、集団の個体数を 300 個体、エリート保存する個体を 4 個体、突然変異率は 3% とした。

生成された 8 小節の曲の楽譜を図 7、16 小節の曲を図 8 に示す。図 7 に示す 8 小節の曲では、「起承転結」の構造を持ち、自然で面白味のある曲が生成できたと考えられる。一方、図 8 に示す 16 小節の曲では、「起承転結」の構造を持つ自然な曲が生成できたが、生成された 8 小節の曲に比べると曲の展開や構造が分かりづらい、という問題があると考えられる。これは、本手法で用いた評価関数だけでは小節数が増え、曲が長くなった場合には不十分であったためではないかと考えられる。

## 5 むすび

本研究では、遺伝的アルゴリズム (GA) を用いて、自動作曲を行う手法を提案した。提案手法では、ある程度短めの曲を生成することを目的とした場合、構造が分かりやすく自然な曲を生成することができた。しかし、長い曲を生成することを目的とすると、構造が分かりやすく自然な曲が生成されにくくなる。

今後の課題としては、小節数を長く設定しても優れた曲を生成できるよう、評価方法を吟味するということが挙げられる。また、ユーザから曲のモチーフや、「起承転結」だけに限らない曲の構造など何らかの入力を受け付け、それを反映した曲を生成することが考えられる。さらに自動的に生成された曲に対してユーザがその曲中の気に入った旋律や気に入らなかった旋律などを入力し、それをデータとして蓄積することによってシステムが学習していくといった作曲支援システムとしての応用が考えられる。

## 参考文献

- [1] 今井 繁, 長尾 智: “遺伝的アルゴリズムを用いた自動作曲”, 電子情報通信学会技術研究報告, AI98-9, pp.59-65, 1998.
- [2] 山田 拓志, 椎塚 久雄: “遺伝的アルゴリズムを用いた自動作曲について”, 情報処理学会音楽情報科学研究会報告, 27-2, pp.7-14, 1998.
- [3] 梅本 あずさ, 内山 幹乃扶, 河合 敦夫, 椎野 努: “音楽理論と経験的知識を統合活用した作曲支援システム”, 情報処理学会音楽情報科学研究会報告, 23-11, pp.57-62, 1997.
- [4] Ames, C: “The Markov process as a compositional model: a survey and tutorial” Leonardo 22(2), pp.175-188, 1989.
- [5] Curtis Roads: “THE COMPUTER MUSIC TUTORIAL”, 東京電機大学出版局, pp.718-727, 2001.