Vol.2010-MPS-79 No.11
2010/7/12

遺伝的プログラミングによる音楽生成モデル

小松恭子[†1]　山中朋美[†1]
高田雅美[†1]　城和貴[†1]

本稿では音楽生成モデルを提案する．このモデルはコード進行生成モデルとメロディ生成モデルからなり，それぞれ遺伝的プログラミングを適用する．まず部分コード進行を生成し，隠れマルコフモデルの特徴を利用して、得られる部分コード進行に適したメロディ列を作成する．そしてメロディ列を初期個体とし，部分メロディ生成を行う．遺伝的プログラミングを用いることで，メロディにおける音価の制限をなくし，あらゆる音の表現が可能となる．

A Music Composition Model with Genetic Programming

Kyoko Komatsu ,[†1] Tomomi Yamanaka ,[†1]
Masami Takata [†1] and Kazuki Joe [†1]

In this paper, we propose a music composition model. This model is constructed two parts, which are code progression and melody. Both parts are adopted Genetic Programming(GP). In the first part, a partial code progression is generated. Then, suitable melody lines related to the partial code progression is led by Hidden Markov Model. In the second part, a partial melody is generated by GP, in which initial individuals are based on the suitable melody lines. By using GP, since it doesn ' t limit phonetic value, every note can be expressed.

†1 奈良女子大学大学院人間文化研究科
Graduate School of Humanities and Sciences, Nara Women's University

1. はじめに

近年，ウェブサイトコンテンツや映像作品に使用する BGM など，任意のテーマに添い，かつ著作権フリーな音楽素材への需要が高まってきている．これを解決する方法の 1 つに，作曲システムがある．一般ユーザが作曲できるシステムの普及により，各自のホームページに音楽を付随させることが容易になった．作曲システムには，曲調・テンポ・感情などのパラメータを与えるだけで曲が生成されるものから，対話的に，つまり音楽生成過程で何度も自分の好みを選択して，ユーザの使用用途に合う曲を生成できるものまである．しかし，音楽には専門的な要素が多く，表現する人の豊富な音楽的知識と経験から音楽は生まれる．このため，音楽初心者が作曲活動に取り組むことは難しい．そこで，これらの問題を回避するために，自動で音楽を作曲するシステムが提案されている[1)–3)]．これらのシステムでは，遺伝子オペレータが適用されているが，遺伝的アルゴリズム（Genetic Algorithm : GA）[4)] を用いているため，1 音の長さに対する制限がある．また，ユーザの意見を反映させるための対話がユーザへの負担の増加に繋がる．さらに，簡単な音楽的制約によって音楽性を保つようにしているが必ずしも生成される楽曲が音楽的であるとは言えない．このように，作曲自体を 1 つの問題として扱うことはあまりにも複雑であり，音楽生成や自動作曲を実現することは容易なことではない．そこで本研究では，遺伝的プログラミング (Genetic Programming : GP)[5),6)] を適用することで，音の長さに制限をなくし，和音のもつ機能を利用して音楽性を保つ音楽を作曲するための音楽生成モデルを提案する．このモデルは，コード進行を生成するモデルとメロディを生成するモデルで構成される．

2 章で既存のシステムとその問題点について述べ，3 章で提案モデルについて説明する．そして 4 章で実験と実験結果を示し，5 章でまとめと今後の課題を述べる．

2. 既存のシステムとその問題点

本章では，GA[4)] を用いた既存のシステムとして 3 つ紹介する．

まず 1 つ目に，Jazz 即興演奏システムとして，GenJam[1)] がある．これは，学習・改良・デモの 3 つのモードを備えており，GA を用いて作曲を行う Macintosh 専用システムである．GenJam は小節個体とフレーズ個体を用意し，初期個体は予め音楽作成ソフトで得た曲から生成している．学習モードでは，個体ごとの適応度を確立することを目的とし，改良モードでは，交叉・突然変異・逆位を繰り返し行うことで，個体を発展させる．音個体は，GA に適用するため，配列で表されたビット列となっている．曲を生成するにあたり，

休符・音・長さの情報が必要になるので，休符を 0，音符を 1〜14，継続を 15 としている．GA を適用するにあたり，配列の最小単位を 8 分音符としているため，この配列要素を組み合わせることによって 8 分音符以上の長さの音符を表現することは可能であるが，それよりも短い音や 3 連符などを表現することはできない．また，選べる音が限られた 14 音しかないため，作曲できる曲の幅が一般的な曲に比べて狭い．

2 つ目のシステムとして，作りたい曲のイメージや希望するコード進行の並びを入力データとし，自動的に作曲を行うものがある[2]．このシステムは，経験データ蓄積システム・コード進行作成システム・メロディ作成システムの 3 つから構成される．経験データ蓄積システムでは，過去の音楽からコード進行をサンプルデータとして保存し，開発者の経験情報に基づきネットワーク状に蓄積する．コード進行およびメロディ作成には，GA を用いる．コード進行生成では，入力されたコードを基にコード進行と各コードの長さを決定する．メロディ生成では，1 コード単位に対して GA を適用する．1 つのコード遺伝子で 1 小節分を表すため，コードは常に 1 小節単位でしか変化しない．メロディ遺伝子は全体で 1 小節分を表し，1 音は 16 分音符を表す．また別に，発音するかしないかを表す起伏データを用いることで，16 分音符より長い音の表現が可能となる．しかし，より短い音や 3 連符を表すことができない．

3 つ目として，自然な曲の生成のため，1 曲を起承転結の流れで表すシステムがある[3]．これは，既存の曲や対話的に作成した楽曲複数を起承転結に分別した状態でデータベースに用意し，その中に含まれる曲のリズムと音高パターンの最適な組み合わせを見つける最適化問題とみなして GA を用いているものである．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生成されるが，データベースに存在しないメロディを作ることはない．

3. 提案モデル

3.1 概　　要

本研究では音楽の専門知識を必要としない音楽生成モデルを提案する．本モデルではコード進行を予め決めてメロディを当てはめるという方法を用いて音楽生成を行う．なお，本稿で提案するモデルにおいて，テンポや強弱は一定なものとして扱う．

図 1 にモデルの全体像を示す．このモデルは，コード進行生成モデルとメロディ生成モデルの 2 つのモデルで構成される．既存の音楽に拘束されない音楽生成を行うため，両モデルに遺伝子オペレータを用いる．遺伝子オペレータとして，GA と GP がある．短い音や 3 連音符・付点音符などのあらゆる音の長さを個体で表現するために，配列をデータ型とする

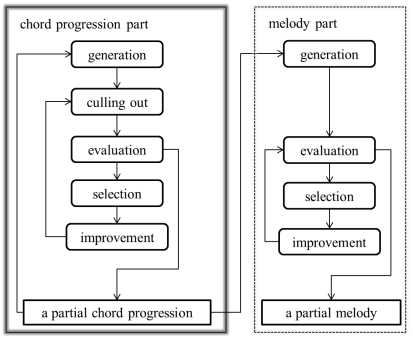


図 1　general representation of the model

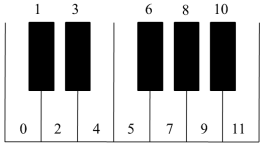


図 2　digitalization of notes

GA ではなく木構造をデータ型とする GP を使用して複雑な構造を表現する．

音楽は複数の小節から成り立っている．コード進行生成モデルでは小節単位で部分コード進行を順次完成させ，最終的に最良の個体を 1 つ出力する．

メロディ生成モデルでは，この 1 個体に合う部分メロディを生成する．まず初めに隠れマルコフモデル（Hidden Markov Model : HMM）[7] を使って，得られる部分コード進行に適したメロディ列を作成する．その結果をもとに GP の初期個体群を生成する．これにより，適当に割り当てたメロディよりもコードにあったメロディを初期個体とすることが可能となる．最終的に，部分メロディとして出力される結果は 1 個体とする．

評価部分は両モデル共に，マルチエージェント[8] の機能を用い，多数のエージェントにそれぞれ異なった視点からの評価をさせることで，音楽を多角的に評価する．

交叉においては，階層と葉の情報が一致する部分木を交換しても，同じ個体しか生成されないため，好ましくない．そこで，交叉させる部分木を比較し，同一の部分木であるならば，交叉点を選びなおす．

3.2 個体の情報

3.2.1 コード進行のための個体

1 つの木で 1 つのコード進行を表す．コード進行として必要な情報はコードとそのコードの長さ（音価）である．

図 2 は，鍵盤上の音を数値化した例である．コードは 1 つの和音形態に 12 種類ある．和音形態は**表 1** に示すように 17 種類ある．このため，ノードは 204（17*12）種類考えられる．**表 2** はノードの 1 例である．構造決定には，必要最小限のデータ量で 1 つの構造を表

Vol.2010-MPS-79 No.11
2010/7/12

表 1 kind of chord configuration

chord configuration	signage
Major triad	$\square$
Minor triad	$\square_m$
Dominant seventh	$\square_7$
Minor seventh	$\square_{m7}$
Major seventh	$\square_{M7}$
Minor Major seventh	$\square_{mM7}$
Diminished seventh	$\square_{dim}$
Augmented triad	$\square_{aug}$
Diminished triad	$\square_m^{-5}$
Augmented seventh	$\square_7^{+5}$
Dominant seventh Flatted fifth	$\square_7^{-5}$
Augmented Major seventh	$\square_{M7}^{+5}$
Minor seventh Flatted fifth	$\square_{m7}^{-5}$
Major sixth	$\square_6$
Minor sixth	$\square_{m6}$
Suspended fourth	$\square_{sus4}$
Dominant seventh suspended fourth	$\square_{7sus4}$

表 2 example of node

node	major triad	minor triad
C	047	037
Cis/Des	158	148
D	269	259
Dis/Es	37 10	36 10
E	48 11	47 11
F	590	580
Fis/Ges	:	:
G	:	:
Gis/As	:	:
A	:	:
Ais/B	:	:
H	11 36	11 26

すことにより計算量を減らすことかつあらゆる進行パターンを生成できることが必要である．本提案モデルでは，各ノードにコードを割り当てる．そして木構造の階層はコードの長さを表す．木構造の階層とコードの長さを対応させるために，根の音価を一定の長さを表すものとする．2 分木の場合，下の階層は上の階層の長さの半分になるものと定義する．2 分木以外の木を用いる場合，2 の倍数の分木であれば 2 分木に置き換えることが可能である．また，それ以外の場合は一般的な音楽のコード進行では考えられない．ゆえに，コード進行のための構造は，2 分木とする．コード進行は，深さ探索で得られる葉の順番で表される．また，そのコードの音価は葉が属する階層の深さで表される．これにより，コードとそのコードの長さを同時に表せる木構造となる． コード進行においては，音価はそれなりの長さを必要とするため提案する木構造では，大きな階層構造を必要としない．そのため，メモリの容量の負荷を削減でき，探索領域も少なく表すことができる．

3.2.2 メロディのための初期個体

図 3 は，メロディを表すための木構造の 1 例である．メロディを表すための木構造は，各ノードに音を割り当てる．3.2.1 項と同様に，深さで音価を表すものとする．分木の数は，2 または 3 とする．これにより，3 連音符の表現を可能にする．ノードの情報は，図 2 のような鍵盤音 12 音と休符（*）と継続（〜）の 14 種類使用する．ただし，鍵盤音については 1

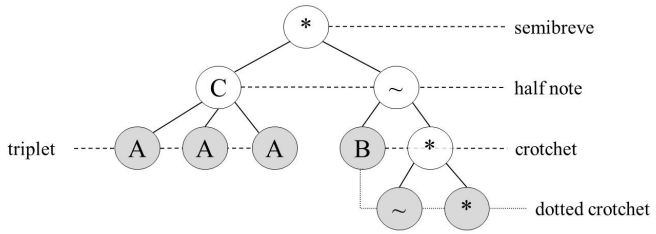


図 3 representation of triplet and dotted crotchet

表 3 note behavior

	T	D	S
primary triad	I	V(7)	IV
secondary triad	VI	VII	II

オクターブに限らず用意する．このため，ノードは 12×（オクターブ分）＋ 2 種類存在する．継続は前のノードが表す音を継続するものである．これにより，付点音符や次の小節にまたがる音符を表現できる．

GP において，初期個体の設定は重要である．なぜならば，初期個体の状態によって，収束速度や収束値が変化する可能性があるからである．本項で対象とするメロディは，先に生成されたコード進行を基に構成されるべきである．そのため，ランダムに初期個体を発生させると，コード進行に合わない個体が大量にできる可能性がある．この問題を解決するために，まず，HMM を使用して，基となる個体を一意的に作成する．HMM は，パラメータが未知のマルコフ過程であると仮定し，観測可能な情報から，その未知のパラメータを推定するというものである．モデルのパラメータを推定する Baum-Welch アルゴリズム[7] と HMM の最尤状態遷移列を求める Viterbi アルゴリズム[7] を用いて，コード進行モデルから得られた部分コード進行に最も尤もらしいメロディ列を求める．例えば部分コード進行 CCFCDCAC を与えた場合，メロディ列 DDCEA*CEA が一意的に決定される．次に，HMM で一意的に決定されたメロディ列に似通った個体群を GP で生成する．

3.3 淘　　汰

音楽には，すべてのジャンルにおいて，最低限必要とされている性質がある[9]．それは，曲の基となる音や始まり・終わりの音になる Tonic（T），T に進行しやすい音の Dominant（D），経過・つなぎに使われ，D に進行しやすい音である Subdominant（S）の 3 つを指

表 4 fundamental note in each key

major	I	II	III	IV	V	VI	VII
C :	0	2	4	5	7	9	11
Cis :/Des :	1	3	5	6	8	10	0
D :	2	4	6	7	9	11	1
…	…	…	…	…	…	…	…
H :	11	1	3	4	6	8	10

minor	I	II	III	IV	V	VI	VII
c :	0	2	3	5	7	8	10
cis :/des :	1	3	4	6	8	9	11
d :	2	4	5	7	9	10	0
…	…	…	…	…	…	…	…
h :	11	1	2	4	6	7	9

す．そこで，コード生成では，この性質に適さない個体を淘汰する．これより，コード進行の流れは，T → S → D → T が基本である．**表 3** は音階を I・II・III・IV・V・VI・VII で表した時の対応表を表す．III の音はこの性質に当てはまらないので表記しない．**表 4** は長調・短調それぞれのコード C〜B における音階 I〜VII の和音の根音を表す．長調は大文字 C〜H で表し，短調は小文字 c〜h で表している．両調において和音形態により表される和音は異なるが根音はすべて一致する．

遺伝子操作において，必ずしも T → S → D → T に適する個体ばかりが存在するわけではない．しかし，これに適さない個体を全て淘汰すると，世代内の多様性が維持できない．そこで，D → S となる葉をもつ個体のみ淘汰する．

また，コード進行におけるコードの変化は，一般的に最短で 4 分音符の長さであるといえる．このため，階層の深さがそれ以上の音価を表す個体も淘汰する．

4. 実験と結果

4.1 コード進行生成

4.1.1 実 験 環 境

ハ長調の曲のためのコード進行生成の実験を行う．パラメータは，個体数 nI，交叉率 c，突然変異率 m，最大世代数 G がある．今回は nI=5，10，50，60，70，80，90，100，c=0.8，m=0.1，G=50 とする．

本実験段階では，マルチエージェントによる評価のモデルが確立されていない．そこで，適応度 $fitness$ を計算するために，**表 5** のコードの割合と 1 個体におけるコード変化数の

表 5 basic chord in each key

chord	M 1st	M 4th	M 5th	m 1st	m 4th	m 5th	M 3rd
Major/minor	T	S	D	T	S	D	
B/g♯	B	E	F♯m	G♯m	C♯m	D♯m	D♯
E/c♯	E	A	B	C♯m	F♯m	G♯m	G♯
A/f♯	A	D	E	F♯m	Bm	C♯m	C♯
D/b	D	G	A	Bm	Em	F♯ m	F♯
G/e	G	C	D	Em	Am	Bm	B
C/a	C	F	G	Am	Dm	Em	E
F/d	F	B♭	C	Dm	Gm	Am	A
B♭/g	B♭	E♭	F	Gm	Cm	Dm	D
E♭/c	E♭	A♭	B♭	Cm	Fm	Gm	G
A♭ /f	A♭	D♭	E♭	Fm	B♭m	Cm	C
D♭ /b♭	D♭	G♭	A♭	B♭ m	E♭ m	Fm	F
G♭ /e♭	G♭	C♭	D♭	E♭ m	A♭ m	B♭ m	Bv
variation	M7 sus4	M7	7 sus4	m7	m7	M7	7

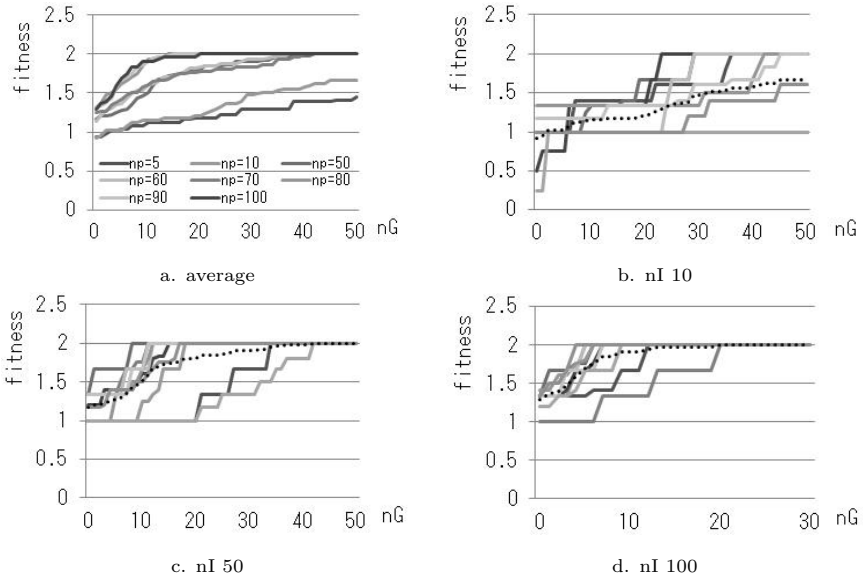
割合に対する加減算を用いる．表 5 は長調（C〜B）と短調（c〜b）が持つ調性による基本コードの例である．調性ごとに頻繁に使われるコードには決まりがある．1 つ目は，調性を保つための計算を行う．本実験では，ハ長調のコード進行を生成することを目的としている．そこで，この 16 個のコードに一致する葉ノード数の合計 $sum1$ を葉ノード数 j で割った数 $sum1/j$ を $fitness1$ に加算する．2 つ目は，コードの進行の仕方に関する計算を行う．淘汰により，すべての個体の階層 $level$ は 4 以下である．$level$=4 では 2 分音符の長さを表すが，コード進行で常に 2 分音符の変化をすることは稀である．よって，コード進行は 1 小節分の長さである $level$=3 を基準とする．そこで，$level$=3 の葉の数 $sum2$ が 1 の場合は $fitness2$ を $1/j$ とし，$2 \leq sum2 \leq 4$ を満たすものは $fitness2$ を 1.0 とするものとする．以上の評価より，

$$fitnss = fitness1 + fitness2 = \begin{cases} \dfrac{sum1 + sum2}{j} & (0 \leq sum2 \leq 1) \\ \dfrac{sum1}{j} + 1 & (2 \leq sum2 \leq 4) \end{cases} \quad (1)$$

となり，$0 \leq fitness \leq 2$ となる．

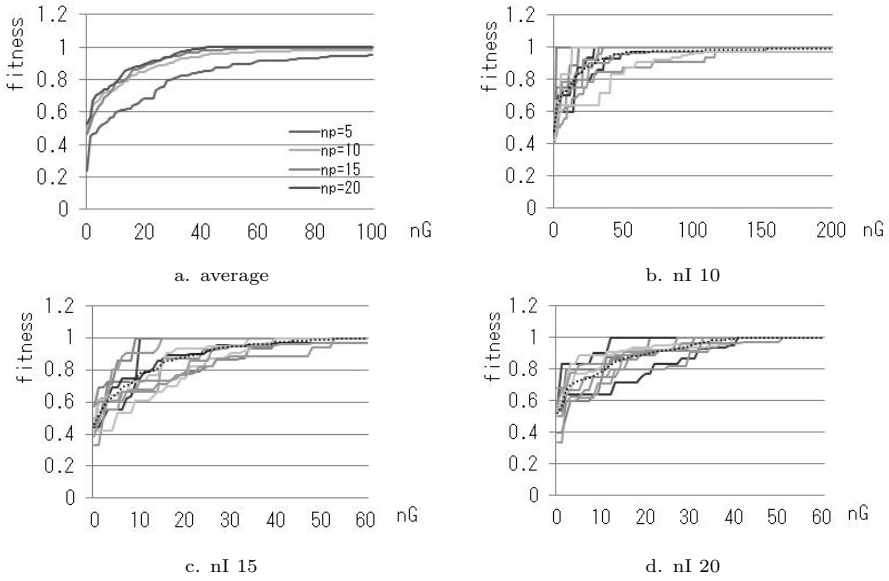
4.1.2 実 験 結 果

図 **4** は各個体数に対する適応度の推移を示したものである．図 4 は nI=5，10，50，60，



a. average　b. nI 10　c. nI 50　d. nI 100

図 4　process of fitness in chord progression generation



a. average　b. nI 10　c. nI 15　d. nI 20

図 5　process of fitness in melody generation

70，80，90，100 の 10 回実験における平均適応度を示し，図 4-b は nI=50 の，図 4-c は nI=60 の，図 4-d は nI=100 の 10 回の実験で得られた適応度の推移をすべてグラフ化したものであり，点線がそれぞれ平均値を示す．

nI=5 では 10 回中 2 回，nI=10 では 10 回中 6 回，最適解へ収束しているが，残りは nG=50 で収束することなく局所解へ陥る．つまり，nI=5，10 のように個体数が少ない場合，コード進行のノードの種類が 204 あるため局所解へ陥りやすいものと考えられる．

nI=50 では 42，nI=60 では 37，nI=70 では 40，nI=80 では 14，nI=90 では 15，nI=100 では 20 世代目までに 10 回とも最適解へ収束する結果となっている．つまり，個体数が多いほど，収束速度が速くなり，nI=80 以上では，ほぼ等しくなる．

4.2　メロディ生成

4.2.1　実 験 環 境

パラメータはコード進行生成と同様に，個体数 nI，交叉率 c，突然変異率 m，最大世代数 G がある．今回は nI=5，10，15，20，c=0.8，m=0.1，G=250 とする．

入力するコード 1 つの長さを 2 分音符に固定する．コード進行同様，マルチエージェントが用いられないため，入力コード進行列の各コードとの整合性と音価において加減算を用いる．1 つ目は，与えられたコード進行に合うメロディの生成率に対する評価である．そのため，与えられたコードに含まれる音であるかどうかによって $fitness$ を計算する．合う音の合計数 $sum3$ を 1 個体の葉ノード数 j で割った値 $sum3/j$ を $fitness$ に加算する．1 音 1 音の音価が長すぎるメロディは退屈な印象を与える．そこで，2 つ目では，1 音の音価が全音符以上の長さを持つ葉の合計数 $sum4$ を j で割った値 $sum4/j$ を $fitness$ から減算する．以上の評価を用いると，

$$fitness = \frac{sum3 - sum4}{j} \qquad (0 \leq sum3, sum4 \leq j, 0 < j) \tag{2}$$

となり，$-1 \leq fitness \leq 1$ となる．

4.2.2　実 験 結 果

入力コード列 CCFCDCAC に対して HMM によるメロディ列 DDCEA*CE が得られる．

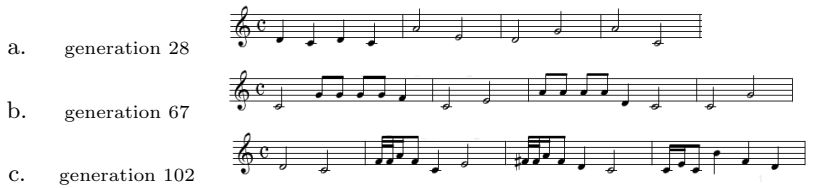


図 **6** generated melody score

図 5 は各個体数に対する適応度の推移を示したものである．図 5-a は nI=5，10，15，20 の 10 回実験における平均適応度を示し，図 5-b は nI=10 の，図 5-c は nI=15 の，図 5-d は nI=20 の 10 回の実験で得られた適応度の推移をすべてグラフ化したものであり，点線がそれぞれの平均値を示す．

nI=5 では，10 回中 6 回は最適解へ平均 43 世代目で収束しているが，4 回は nG=250 で収束せず，133，144，218，244 世代目で局所解に陥る．

nI=10 では，10 回中 7 回は最適解へ平均 41 世代目で収束しているが，3 回は nG=250 でも収束せず，55，70，153 世代目で局所解に陥る．

nI=15 では，10 回中 8 回は最適解へ平均 31 世代目で収束し，2 回は nG=250 で収束せず，41，53 世代目で局所解に陥る．

nI=20 では，51 世代目までに 10 回とも最適解へ収束している．

以上の結果から，個体数が少なければ最適解への収束が遅く，収束する場合の平均世代数を超えても収束しない場合は局所解に陥ることが確認できる．ただし，個体数が 5〜20 個においては個体数に限らず最良の場合，収束するといえる．

また，個体数が多ければ速く収束し，少なければ最良の場合においても収束速度は遅い．したがって，この実験から，個体数は多いほど最適解に辿り着きやすいと考えられる．

この実験で生成されたメロディ例を**図 6** にあげる．これらは nI=20 での結果であり，図 6-a，図 6-b，図 6-c はそれぞれ 28，67，102 世代目で収束したものである．個体数によらず，収束速度が遅いものほど，細かい音が出力される傾向があるといえる．

5. まとめと今後の課題

本稿では，GP を適用することによって，自動で作曲するための音楽生成モデルを提案した．GP 特有のパラメータを指定することで，コード進行とメロディを自動生成する．メロディ生成では，HMM によりコード進行列に最適なメロディ列を決定し，初期個体とした．木構造の階層を音価として捉えることで，より細かい音の表現が可能になり，さらに 2 分木と 3 分木を用いることで，3 連音符や付点音符などのあらゆるリズムパターンが生成できる．また，HMM の状態遷移推移率を決定するための既存の曲を変更することで，パラメータの設定が同じでも，雰囲気の違う音楽を生成できると考えられる．

本稿では，ハ長調の音楽生成を目的とした実験を行った．個体数に限らず，最適解への収束が速いものは，比較的音価の長いメロディが，反対に，収束が遅いものは，比較的音価の短いメロディが得られる傾向になった．しかし，今回の評価ではコード進行列のパターンやメロディラインに使用する音の制限が強いため，評価方法を重点的に見直す必要がある．

今後，マルチエージェントシステムの評価により音楽性を保ち，さまざまなパラメータの選択によりオリジナルな音楽の生成を目指す．これにより作成されるシステムでは，著作権フリーで，音楽初心者を含むシステム利用者が，個人による作曲時間に比べて短時間で，使用用途にあう音楽データを得ることができると期待される．さらに，Jazz に代表されるような即興演奏に対応していきたいと考えている．

参 考 文 献

1) John,A,Biles.:GenJam: A Genetic Algorithm for Generating Jazz Solos,In Proceedings of the 1994 International Computer Music Conference, ICMA, SanFrancisco, pp.3-4 (1994).
2) Yamada,T. and Shiizuka,H.:Automatic Composition by Generic Algorithm, IPSJ Special Interest Group on Music and Computer 27-2, pp.7-14 (1998).
3) Tanaka,T., Toyama,F. and Shoji,K.:Automatic Composition by Combination of Pitch Transition Patterns and Rhythm Using a Genetic Algorithm, IPSJ Special Interest Group on Music and Computer 41-8, pp.43-48 (2001).
4) J,H,Holand.:Adaption in Natural and Artificial Systems, University of Michigan Press, MIT Press (1992)
5) John,R,Koza.:Genetic Programming: on the programming of computers by means of natural selection, MIT Press (1992)
6) John,R,Koza.:Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press (1994)
7) Paul,Taylor.:Text-to-Speech Synthesis, Cambridge University Press (2009).
8) Adelinde,M,Uhrmacher. and Danny,Weyns.:Multi-Agent Systems: Simulation and Applications (Computational Analysis, Synthesis, and Design of Dynamic Models Series), Crc Pr I Lic (2009).
9) William,E,Caplin.:Classical form: a theory of formal functions for the instrumental music of Haydn, Mozart, and Beethoven, Oxford University Press (1998).