Vol.2010-MUS-88 No.11
2010/12/5

好みのバイアスの学習を行う作曲モデルを用いた作曲支援システムの実現

蓮井 洋志　室蘭工業大学

本研究では、作曲モデルにメロディー内のユーザの好みの偏りを学習する方法を導入し、それを用いて対話型選択集団山登り法を利用した対話型作曲支援システム rank-c-Sonneteer2 を実現する。本手法は、評価のランクによって親を選択する集団山登り法である。Rank-c-Sonneteer2 は、対話、親の入れ換え、選択、学習と生成をくり返す。まず、対話において、ユーザが、入力のメロディーを改善、評価する。評価では各音符ごとに 4 段階評価する。親の入れ換えにおいて、評価値が親よりも高ければ、前の親と入れ換えて、親ベースに登録する。選択において、システムは評価のランクに親を選ぶ。学習と生成において、作曲モデルがシステムに選ばれた親を 11 回学習し、親ベースの他の親を 1 回学習し、その学習情報をもとに近傍解、つまり子を生成する。この 4 つのプロセスを好みのメロディーを作曲するまで繰り返す。モデルは学習データをもとにメロディーを生成するが、ランダムウォークで生成するために好みの部分を持たないメロディーを作曲してしまう場合がある。モデルが音符ごとの 4 段階の評価に対して好みの部分の音符の重みを大きくして学習すれば、よりユーザの好みを反映した近傍解を生成できる。本論文では、このモデルを実現したシステムとそのシステムを使って作曲したメロディーについて述べる。

Computer Aided Composition System with Composing Model to Learn Favor Bias

Hiroshi Hasui　Muroran Institute of Technology

In this study, I implemented the computer aided composition system, rank-c-Sonneteer2, with the interactive selective population climbing, where the composing model introduced on learning the favorite bias of the user's evaluation. This method is population climbing which selects the parent with respect of the rank of its evaluation value. Rank-c-Sonneteer2 repeats the interaction, the exchanging the parent, the selection, and the learning and generating until the user gets the favorite melody. First, in the interaction, the user improves the inputted 10 melodies into more favorite melodies, and evaluates the improved melodies. The evaluation method is that the user evaluates 4 level of every note in these melodies. In the exchanging the parent, the system exchanges the offspring higher evaluation value with the parent, and enters into the parent base. In the selection, the system selects 10 parents with respect of the rank. In the learning and generating, the composing model learns the selected melodies 11 times and the other parents in the parent base once, and generates the melody, the offspring near the parent, with randomwalk. Although the model generates the melody based on learning information, it sometimes generates a melody which does not have favoroite part, because of generating with randomwalk. If can learn the favorite bias, the model can generate neighbours which are more reflected on the user's favor. In this paper, we described about the implementation of rank-c-Sonneteer2 with this composing model and the composed melodies with the system.

1. はじめに

私は対話型作曲支援システムを作成してきた。このシステムは、既存の商用製品のような音符を入力するためのインタフェースを持つだけでなく、作曲のアイデアを与えるために自動作曲システムや作曲モデルなどを援用して、作曲を行なう。作曲の発想支援をともなったシステムである。

私は、対話型遺伝的アルゴリズムを用いた i-Sonneteer を実現した[1)–6)]。このシステムはユーザがシステムを用いて好みのメロディーを作曲することを目標とする。このシステムは、個体として作曲モデルを用い、メロディーを遺伝子を用いる。作曲モデルは 7 個の遺伝子を学習し、その情報をもとにメロディーを生成する。7 個の遺伝子は、メロディーベースから遺伝的操作を用いて選択する。ユーザはモデルの生成したメロディーの好みの度合いを評価し、それを好みに合わせて改善する。評価値は生成した個体の適応度となる。改善行為を繰り返すうちにユーザが自分の好みのメロディー像に気付き、最終的に、ユーザが改善したメロディーを作曲モデルが学習するために、作曲モデルが好みのメロディーを作曲する。しかし、このシステムは改善にかかる手間が多すぎた。

そこで、i-Sonneteer を改良した方法として対話型選択集団山登り法[7)] の rank-c-Sonneteer[8)] を実現した。このシステムは、[**対話**]、[**親の入れ換え**]、[**選択**]、[**学習と生成**] をくり返す。[**対話**] でユーザがシステムの作曲したメロディーを改善し、評価する。[**親の入れ換え**] でユーザが改善したメロディーが親のメロディーの評価値以上の評価をうけたときに、システムが親と子を入れ換える。そして、それらの子を親の代わりに親ベースに入れる。[**選択**] で親ベースの中で評価値の高いほうから近傍解を生成するための親を選択する。[**学習と生成**] で作曲モデルが、選択された親を重みを均一にして 11 回、親ベースのそれ以外の親を 1 回学習し、その学習結果をもとに子をランダムウォークで生成する。そのため、

子は親の近傍解となり、山登り型の探索となる。

i-Sonneteer では交叉のために、大きく学習するメロディーが入れ替わる。そのため、前の世代で好みに感じていたメロディーがあっても次世代には遺伝しない場合が多かった。反面、rank-c-Sonneteer は親の近傍解を次世代に残す。そのために、作曲がやりやすくなった。

しかし、rank-c-Sonneteer は、親を 11 回学習するためにその近傍のメロディーを生成することができたが、モデルの生成するメロディーの中に前の世代で存在していた好みの部分がない場合があった。

そこで、メロディーの全音符の 4 段階評価を行ない、ユーザの好みの部分、好みでない部分を明確に分ける。作曲モデルは、好みの部分の重みは大きく、好みでない部分の重みは小さくなるように学習するようにした。このモデルを用いたシステムを rank-c-Sonneteer2 と呼ぶ。これによって、ユーザの好みの部分が子に残りやすくなり、より作曲しやすくなることが期待できる。

また、作曲モデルの学習方法にスムージングを用いることで、ユーザの好みと思われる音高を生成しやすくした。

2 節では、関連文献について述べる。3 節では対話型選択山登り法を提案し、4 節ではシステム構成を述べ、どう実現したかを述べる。5 節でシステムを使って試験的に作曲を行い、それについて考察し、6 節でまとめる。

2. 関 連 文 献

対話型作曲システムとしては、着メロの作曲を目的とした文献 9)–11) が有名である。これは、対話型遺伝的アルゴリズムを利用したシステムで、遺伝子として音符を用いる。音楽理論を用いることでメロディーに制約を設け、探索範囲を狭くし、ユーザの評価中心に作曲する。好みのフレーズを Virus とすることで、好みのフレーズをくり返したり、子孫に遺伝させたりする。

無調性音楽を作曲するシステムとしては、文献 12)，13) が有名である。対話的な遺伝的プログラミングで実現している。曲中の全音符を関数型プログラムで表現したものを染色体として用いる。交叉における親の選択をユーザがデータベースから選んで行なうなど、確実に作曲の効果を上げるためにユーザの操作を利用する。プロの作曲家がメロディーを作曲することを目標としている。

ユーザの手間が多いことが対話型作曲システムの大きな問題である。文献 14) は、対話型遺伝的アルゴリズムのユーザの手間を減らすためにニューラルネットワークがユーザが好みであると判断したメロディーを学習し、それを評価基準としてメロディーを 100 世代自動で進化させるシステムを開発した。これは、100 世代自動で進化した後、またユーザがメロディーを評価する。収束速度が速くなり、手間は格段にへった。

メロディーの断片をミュージックブロックとし、ミュージックブロックを遺伝子として、それらを交叉、突然変異させることでメロディーを再合成するのは、文献 15) のシステムである。生成したメロディーの適応度評価はユーザが行なう。これは一つの対話型進化的計算手法を用いたシステムである。

Jazz の即興演奏において、演奏者の相手の演奏者をつとめるシステムは文献 16) である。良い演奏をした部分は良いとコンピュータにメッセージを送り、悪い演奏の部分は悪いとコンピュータにメッセージを送る。その情報をもとに対話型遺伝的アルゴリズムを用いて、即興演奏を行なう。悪い部分の音は演奏するスケールから外し、良い部分の音はスケールに入れるなどの操作を行なう。

リアルタイム作曲を目標としてかかげたものとしては、文献 17)–21) がある。文献 17) はリズムの各ドラムに一つの演奏エージェントを対応させて演奏する。演奏エージェントごとの対話、指揮エージェントとの対話、ユーザの送る教師信号の学習によって演奏を変える。文献 21) はリズムのみならず、伴奏、メロディーもリアルタイム作曲することを目標としたシステムである。現時点ではコードにこだわらない和音の進行を生成できる。

私は、ユーザが好みのメロディーを作曲するのを発想支援する対話型作曲支援システム 1)–8) を実現してきた。これらはユーザの改善行為主導の対話型のシステムで、ユーザが計算機の作曲した曲を評価、改善し、モデルがそのメロディーを学習することで好みのアイデアを持つ曲を生成する。ユーザは、その中にアイデアを発見し、好みのメロディーを作曲する。

私のシステムは複数のモデルを持つ。そのために、メロディーの多様性を許容したまま作曲をつづけられる。多様性が多いと、手間は多くかかるが、その間にユーザの好みがより明確になるために、より好みのメロディーを作曲できる。

3. Rank-c-Sonneteer2 の設計

3.1 Rank-c-Sonneteer2 の対象とする音楽

音律は平均律を用いる。楽曲形式として拍子は 3 パターン、伴奏パターンは 11 パターン用意した。主に、クラシック、J-POP、J-ROCK などである。コード進行は 4 分の 4 拍子 8 小節のものが 46 パターン、3 分の 4 拍子 16 節のものが 2 パターン、12 分の 8 拍子 8 小節のものは 1 パターンを用意した。メロディーの最低音高は楽曲形式によって定まる。そこから 2 オクターブの音域でメロディーを作る。作曲前にその中からそれぞれを選択する。

メロディーは音の長さの列であるリズムと音高列で表される。対応する音の長さと音高を合わせたものを音符とした。ベロシティは小節の第 1 番目の音符だけ強拍となるようにした。音長は基準音長を決定し、その整数倍の長さの音符だけを使う。基準音長、テンポ、音色は楽曲形式によって変更できるようにした。

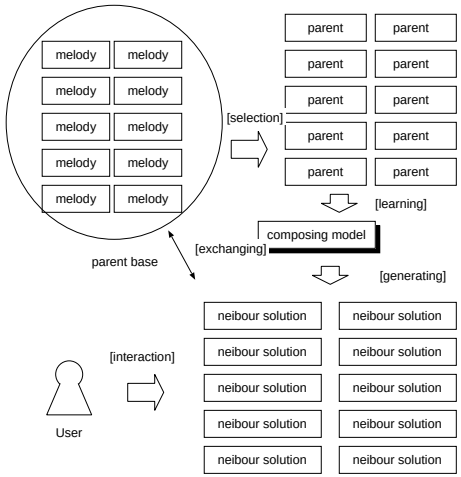

図 1 対話型選択集団山登り法の流れ図

コード進行法で作曲すると、音高はコード構成音を中心に作曲する。コード外音はコードスケールの音だけが許される。しかし、一般の楽曲ではコードスケール以外の音もメロディーの中にある。Rank-c-Sonneteer2 ではコードスケール外音が入ったものも考える。

メロディーの音高やリズムは人によって好みがある。機械的に作曲した場合、好みのようには定めることはできない。ユーザは、音高やリズムはユーザの好みに合うように改善する。

3.2 対話型選択集団山登り法

対話型選択集団山登り法が好みのメロディーを作る処理のプロセスを図 1 に示す。[**対話**]、[**親の入れ換え**]、[**選択**]、[**学習と生成**] の処理を繰りかえすことで、少しずつ好みにあったメロディーにする。

[**対話**]

ユーザは親の近傍解である子のメロディーをよりユーザの好みに合うように改善して、1 音符単位で 4 段階評価をする。メロディー全体の評価は全音符の評価値の合計を正規化したものである。

Rank-c-Sonneteer2 は好みのメロディーを探索するために、好みの度合いを基準に改善した曲を評価する。評価は GUI インタフェースで行なう。

[**親の入れ換え**]

もし、作曲モデルが作曲した子をユーザが編集、評価したメロディーが、その親の評価値より高い値であったら、rank-c-Sonneteer2 は親と子を入れ換える。親よりも評価値が高い子が 2 個以上ある場合は、8-10 番目に評価値の高い親と入れ換える。親は親ベースに登録する。

[**選択**]

Rank-c-Sonneteer2 は 10 個の作曲モデル各々に、親ベースから 1 つ親を選択する。親ベースから、3 つの作曲モデルが 1 番目の評価の解を、2 つの作曲モデルが 2 番目の評価の解を、各々1 つの作曲モデルが 3、4、5、6、7 番目の評価の解を選択する。評価値が同じ時には番号が若いものを優先した。

[**学習と生成**]

作曲モデルが選択した親を 11 回、それ以外の親ベースの親を 1 回学習したあと、ランダムウォークで子のメロディーを生成する。親の学習回数が多いために、子は親の近傍の解となる。

この後、処理は [**対話**] に戻る。

[**対話**] において、モデルが作曲したメロディーか、ユーザが修正したメロディーに満足できたときに、rank-c-Sonneteer2 は終了する。

3.3 自動作曲システム Sonneteer

自動作曲システム Sonneteer は、コード進行法を用いて調性音楽を作曲する。リズムは 4 分音符、8 分音符の数、シンコペーションの数、休符の数から遺伝的アルゴリズムによって作成する。

コード進行法はコード構成音をもとにメロディーの音高を決定する作曲方法のことをいう。コードとは伴奏に使えそうな和音のことをいう。和音は 3 つ以上の音高からなり、それはコード構成音と呼ばれる。コードは時系列に変化する。コードの推移によって人間は音楽の物語性を獲得する。

あるコードに決定されている小節では、コード構成音は協和音となるために良く用いられる。コードスケール音は曲調を変化させるために用いられる。それ以外の音は不協和音となるために Sonneteer では扱わない。

Rank-c-Sonneteer2 は Sonneteer の作曲するメロディーを入力として用いる。

3.4 Rank-c-Sonneteer2 の作曲モデル

作曲モデルは、決定性確率有限状態オートマトン $(\Sigma, Q, S, s_0, \delta, P_{trans}, P_{output}, F)$ である。

Σ : $input$; $start\ time\ of\ note\ or\ rest\ note$

Q : $state$; $\{s_k | k = 0, 1, .., end\}$

S : $symbol\ (pitch)$; $\{p_i | i = 1, .., 25\}$

s_0 : $start\ state\ of\ first\ note$

δ : *state transition function*
P_{trans} : *state transition probability function*
P_{output} : *output symbol probability function*
F : *end state* ; s_{end}

音符の開始時間は状態と対応する。現在の音符の開始時間から次の音符の開始時間が現在の音符の音の長さである。音の長さは、基準音長 D の整数倍である。開始時間 $t \times D$ と状態 s_t が対応づけられる。

状態から状態へ遷移する確率を状態遷移確率 P_{trans} と呼ぶ。その状態において、記号を出力する確率を記号出力確率 P_{output} と呼ぶ。また、コード構成音の出現確率を P_{chord}、非構成音の出現確率を $P_{nonchord}$ と呼ぶ。前の音符の音高と現在の音符の音高の間の音程を i とすると、音程 i の出現する確率を音程出現確率 $P_{interval}(i)$ と呼ぶ。P_{trans} はリズム生成に、P_{output}、$P_{interval}$、P_{chord}、$P_{nonchord}$ は音高生成に使う。

各状態の記号 p_i はその開始時間から始まった音符の音高である。$i = 0$ は最低音高である。p_{12} は最低音高の 1 オクターブ上の音高である。休符は p_{24} と表す。

3.4.1 学習方法

弱起のメロディーを考えないために、初期状態は必ず s_0 である。

メロディーの音休符全部を学習対象とする。入力された音符 *note* が $k \times D$ から始まり、次の音休符の $l \times D$ まで続く場合、状態 s_k から状態 s_l に遷移する。このとき、音符 *note* の長さは $(l-k) \times D$ である。

note の音高が *symbol* であるとき、記号として遷移前の状態に *symbol* を記録する。

複数のメロディーの学習を行なうときには、状態 s_0 から遷移しなおす。つまり、状態は開始時間が早いほうから遅いほうにしか遷移しない。

学習したメロディー全体の k から l に状態を遷移させた音符の重みの合計を $trans_{k,l}$、状態 k の記号 p_k の音符の重みの合計を $output_{k,p_k}$ とする。重みは、好みの度合いを表す。その音符が不自然であるとユーザが評価した場合は 2、自然の場合は 10、好みの場合は 14、特に好みの場合は 18 とした。

状態遷移確率は式 (1) で、記号出力確率は式 (2) で表される。

$$P_{trans}(s_l|s_k) = \frac{trans_{k,l}}{\sum_{l=0}^{end} trans_{k,l}} \tag{1}$$

$$P_{output}(p|s_k) = \frac{output_{k,p} + 1}{\sum_{l=0}^{24}(output_{k,p_l} + 1)} \tag{2}$$

また、$P_{interval}$ は式 (3) のように算出する。ひとつ前の音高と現在の音高の間の音程を i とすると、メロディー中の i の音符の数を $interval_i$ とあらわす。$allnotes$ はメロディー内の全音符数である。

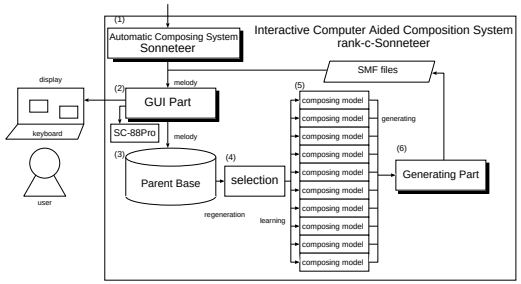

図 2 対話型作曲支援システム rank-c-Sonneteer2 の構成

$$P_{interval}(i) = \frac{interval_i + 1}{allnotes + 25} \tag{3}$$

メロディー内のコード構成音の数を *chord* とすると、以下の式で P_{chord}、$P_{nonchord}$ を表す。

$$P_{chord} = \frac{chord + 1}{allnotes + 2} \tag{4}$$

$$P_{nonchord}(i) = 1 - P_{chord} \tag{5}$$

確率は 0 だと自然なメロディー生成ができない場合があるため、P_{chord}、$P_{interval}(i)$ には音符数に 1 を加えて、$P_{output}(p|s_k)$ には重みに 1 を加えてスムージングした。

3.4.2 生成方法

作曲モデルの生成方法は、オートマトン上をランダムウォークで初期状態から終了状態へ状態遷移確率をもとに遷移する。ランダムウォークで遷移した状態系列をリズムに変換する。例えば、状態遷移系列が s_0、s_2、s_3、s_6 で D の長さが半拍であれば、4 分音符、8 分音符、符点 4 分音符のリズムとなる。

リズムを生成した後で、各音符の音高を決定する。作曲モデルでは、音高を各状態の記号で表す。最初の音符は、開始状態の記号出力確率 P_{output} が最大の音高を選択する。二番目以降の音符は、式 (4) の P_{pitch} が最大の音高を選択する。

$$P_{pitch} = \begin{cases} P_{output}(o_i|s_i) \times P_{chord} \times P_{interval}(p(o_{i-1}) - p(o_i)) \\ \qquad (o_i \ is \ chord \ component) \\ P_{output}(o_i|s_i) \times P_{nonchord} \times P_{interval}(p(o_{i-1}) - p(o_i)) \\ \qquad (o_i \ is \ non-chord \ component) \end{cases} \tag{6}$$

ただし、$p(o_i)$ は音高 o_i の最低音高からの音程の度数である。記号出力確率のほかに、音

程出現確率、コード構成音出現確率、非コード構成音出現確率も音高の選択に関わる。これは、コード進行に沿った自然なメロディーを作るためである。

3.5 再自動作曲機能

自動作曲システムの作曲した 10 曲を入力としているが、嫌いな曲の代わりに新しく自動作曲したメロディーを入れる機能を作った。Rank-c-Sonneteer2 は親ベースの 10 曲だけからしか学習しない。実行中全部のメロディーから突然変異によって選択される可能性のある i-Sonneteer と比較して好みの多様性が少ない。気に入った曲が親ベースにないと、作曲モデルが好みの情報を学習できない。その結果として、発想支援の効果を得ることができない。この機能によって多様性を保持する。

4. Rank-c-Sonneteer2 の実現

4.1 Rank-c-Sonneteer2 のシステム構成

図 2 に対話型作曲支援システム rank-c-Sonneteer2 の構成図を書く。この作曲システムは C++を用いて Linux 上で開発した。Standard MIDI Format(smf) を用いて音楽を表現し、演奏には MIDI 音源装置 SC-88Pro を用いる。デバイスドライバにはフリーウェアの `srgplay-0.80` を用いた。

Rank-c-Sonneteer2 は、(1) 自動作曲システム Sonneteer、(2) GUI 部、(3) 親ベース、(4) 親の選択、(5) 作曲モデル、(6) 楽曲生成部の 6 部品で構成する。(1) で入力するメロディーを 10 曲生成する。(2) で **[対話]**、(3) で **[親の入れ換え]**、(4) で **[選択]**、(5) と (6) で **[学習と生成]** を行なう。

Rank-c-Sonneteer2 は処理を具体的に表す。第 1 世代で、(1) が自動作曲した 10 曲を (2) でユーザが修正、評価し、それを (3) に登録する。(4) で評価値のランクから各々の作曲モデルが一つの親を選択する。(5) で選択した親を 11 回を学習し、それ以外の親 9 曲を 1 回学習する。そのモデルがランダムウォークでメロディーを生成し、(6) でメロディーに伴奏を追加して smf フォーマットのファイルを作る。

第 2 世代以降で、(2) で前の世代で生成したメロディーをユーザが修正、評価し、それらの中で評価値が親より高いものだけ親と入れ替えて (3) に登録する。(4) で評価値のランクから各々の作曲モデルが一つの親を選択する。(5) で選択した親を 11 回を学習し、それ以外の親 9 曲を 1 回学習する。そのモデルがランダムウォークでメロディーを生成し、(6) でメロディーに伴奏を追加して smf フォーマットのファイルを作る。(2) に戻り、繰りかえす。

4.2 対話型インタフェースの実現

図 3 に実行画面を見せる。右側の画面がピアノロールウィンドウで、左側の画面は端末ウィンドウである。ピアノロールウィンドウは、メロディーを表示し、修正および評価ができる。

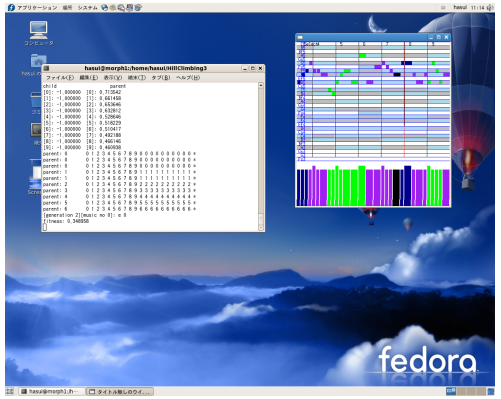

図 3 実行画面

ピアノロールウィンドウは上半分が音高と音長を表し、下半分がその音符のベロシティを表示する。

ピアノロールウィンドウの音の高さが縦軸となり、音の開始時間が横軸となる。コードの構成音は水色で、コードスケール音はクリーム色で音符の背景に表示される。それ以外の音の背景は白色である。音符はそれらの中に表示される。

Rank-c-Sonneteer2 では、1 音符ごとに 4 段階評価を行なう。マウスでピアノロールウィンドウを左ボタンでドラッグすると、その開始地点から終了地点の音符までが選択できる。右ボタンで選択した音符は 1 段階評価が上がる。左ボタンで選択した音符は最低の評価になる。一番下の段階から、不自然、自然、好み、とても好みの順で好みの度合いが大きくなる。音符の色は、不自然な部分の音符は緑色、自然な部分は紫色、好みの部分は紺色、とても好みの部分は黒色である。

メロディー全体の評価値は式 (6) で求まる。i 番目の音符の長さを $length_i$、評価値を $value_i$、最大の評価値を $maxvalue$ とするとメロディー全体の評価値 $eval$ は式 (6) で求まる。

$$eval = \frac{\sum_i length_i \times value_i}{\sum_i length_i \times maxvalue} \tag{7}$$

評価値 $value_i$ は、不自然な音符が 0、自然な音符が 8、好みの音符が 12、とても好みの音符が 16 とした。$maxvalue$ は 16 である。

Vol.2010-MUS-88 No.11
2010/12/5

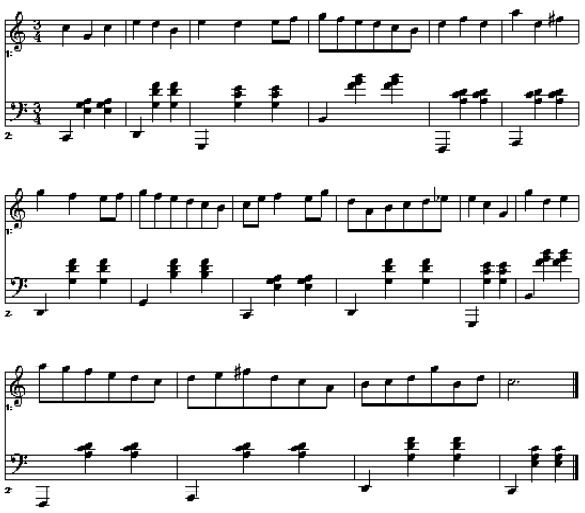

図 4 楽譜 1：ワルツ

1 音符ごとの評価は、ピアノロールウィンドウでマウスをドラッグして行なう。音楽の評価はフレーズで決定される場合が多い。演奏を聞きながら、直感的に好みと感じたフレーズをマウスでドラッグすることは容易である。フレーズのくぎり目はユーザが 1 音符単位で区切ることができる。まれに、フレーズ内の 1 − 3 音符だけおかしいと感じる場合は、その部分を右ボタンでドラッグして不自然にすればすむ。

編集は演奏終了後にキーボードで ‘a’ を押すとできる。マウスをドラッグした部分に音符が移る。‘r’ を押すと修正されたメロディーの再演奏を行える。‘q’ を押すことで編集を終了できる。

5. 実験と考察

このシステムを用いて作曲した結果を図 4 と図 5 に示す。図 4 はショパンのワルツの進行をもとに作成した。伴奏も入っている。第 3 世代で作曲は終了した。図 5 は J-POP のコード進行を学習した N-gram のコード進行生成システムを用いて作成したコード進行で作成した。後者の図には伴奏は入っていない。第 5 世代で作曲は終了した。

2 曲作曲したが、全部で 53 分でできた。評価した曲はすべて改善した。改善前のメロディーと改善後のメロディーの類似度の平均はワルツが 0.4219 で J-POP が 0.5554 であった。類似の計算は以下の式で行った。メロディー i とメロディー j の全音符の開始時間と終了時間が同じものを $duration(i,j)$ 開始時間と音高が同じものを $pitch(i,j)$ とする。メロディー i の全音符数を $all(i)$ とする。

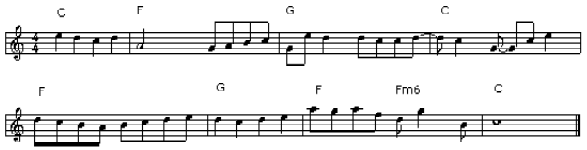

図 5 楽譜 2：J-POP

$$similarity(i,j) = \frac{duration(i,j) + pitch(i,j)}{(all(i) + all(j))} \tag{8}$$

また、再自動作曲機能はワルツが 0 曲、J-POP が 7 曲使った。ワルツは改善行為中心に、J-POP は評価行為中心に作曲した。改善行為を行った曲数自体はワルツも J-POP も多いが、改善した音符数の割合は J-POP のほうが少なかった。このシステムは改善行為の量を減らして作曲できる可能性が高い。

6. ま と め

本論文では、好みの偏りを学習する作曲モデルを用いた対話型選択集団山登り法の対話型作曲システムを実現した。この作曲モデルは生成方法を工夫して好みのアイデアの入った自然なメロディーを生成する。このシステムは比較的ユーザの手間が少なく作曲ができることが期待できる。

改善行為は音楽についての専門的な知識を必要とする。再自動作曲機能を用いて、好みのアイデアを改善行為でなく、評価行為から得るシステムを作ることができれば、音楽に詳しくなくても使用できる。

対話型作曲支援システムは、ユーザがメロディーに改善行為や評価行為の手間をかけるうちに計算機から学び、計算機はユーザの手間の掛けたメロディーから学んでいるうちに両者が歩み寄って作曲ができるのが理想であると考える。手間をかけるうちに音楽の好みを明確にすることができるからである。全音符を 4 段階評価するのは多少手間がかかるが、音楽の好みを深めるために必要である。

今後の課題として、システムとの手間に関する評価、改善行為を多く必要としないシステムへの移行がある。

参 考 文 献

1) 蓮井洋志，小倉久和：対話型進化的手法による作曲システム i-Sonneteer の作成，進化的計算シンポジウム 2007 講演論文集，pp.111–114 (2007).
2) 蓮井洋志，小倉久和：対話型作曲システム i-Sonneteer の他楽曲形式への応用，pp. 157–162 (2008).
3) 蓮井洋志，小倉久和：対話型進化的計算手法による作曲システムにおける HMM の作曲法，情報処理学会第 70 回全国大会講演論文集，Vol.70, No.2, pp.61–62 (2008).
4) 蓮井洋志：N-Best 探索アルゴリズムを利用した k-measure HMM による作曲法，*IPSJ SIG 2008-Music-75*, pp.129–134 (2008).
5) Hasui, H. and Ogura, H.: Optimization of HMM with Interactive Evolutionary Computation in Composing System, *Proc. of 2008 IEEE Conference on Soft Computing in Industrial Application*, pp.171–176 (2008).
6) 蓮井洋志：作曲モデルを用いた対話型作曲支援システムの実現，ファジー知能情報学会論文誌，No.2, pp.247–255 (2009).
7) Hasui, H.: Computer Aided Composition System with Interactive Selective Population Climbing, *Proc. of CSIE 2009*, Vol.5, pp.1–6 (2009).
8) 蓮井洋志：選択的集団山登り法による対話型作曲支援システムの作成，進化的計算シンポジウム 2008，pp.75–78 (2008).
9) 畦原宗之，鬼沢武久：インタラクティブ作曲支援システム〜ユーザの負担の軽減〜，*The 17th Annual Conference of Japanese Society for Artificial Intelligence*, No.1B4-05 (2003).
10) UNEHARA, M. and ONISAWA, T.: Interactive Music Composition System, *Proceedings of SMC 2002* (2002).
11) UNEHARA, M. and ONISAWA, T.: Music composition system with human evaluation as human centered system, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, Vol.7, No.3, pp.167–178 (2003).
12) Ando, D., Dahlsted, P., Nordahl, M.G. and Iba, H.: Computer Aided Composition for Contemporary Classical Music by means of Interactive GP, *The Journal of the Society for Art and Science*, Vol.4, No.2, pp.77–87 (2005).
13) Ando, D. and Iba, H.: Interactive Composition Aid System by means of Tree Representation of Musical Phrase, *Proceedings of CEC 2007*, pp.4258–4262 (2007).
14) Johanson, B.E. and Poli, R.: GP-Music: An Interative Genetic Programming System for Music Generation with Automated Fitness Raters, Technical ReportCSRP-98-13, School of Computer Science, The University of Barmingham (1998).
15) Chen, Y.: Interactive music computation with the CFE framework, *ACM SIGEVOlution*, Vol.2, No.1, pp.9–16 (2007).
16) Biles, J.A.: GenJam: A Genetic Algorithms for Generating Jazz Solos., *Proceedings of the 1994 International Computer Music Conference* (1994).
17) Eigenfeldt, A.: The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine, Vol.5484, pp.498–507 (2009).
18) Eigenfeldt, A.: Kinetic Engine: Toward an Intelligent Improvising Instrument, *In: Proceedings of the Sound and Music Computing Conference* (2006).
19) Eigenfeldt, A.: Drum Circle: Intelligent Agents in Max/MSP, *In: Proceedings of the International Computer Music Conference* (2007).
20) Eigenfeldt, A.: Multiagent Modeling of Complex Rhythmic Interactions in Realtime Performance, *In: Sounds of Artificial Life: Breeding Music with Digital Biology*, A-R Editions (2008).
21) A. Eigenfeldt, P. P.: The Realtime Generative Music System using Autonomous Melody, Harmony, and Rhythm Agents, *12th Generative Art Conference GA2009*, pp.67–76 (2009).