

遺伝的アルゴリズムを用いた自動作曲について

山田 拓志 椎塚 久雄
工学院大学 電子工学科
shiizuka@sin.cc.kogakuin.ac.jp

絵や音楽は、コンピュータの限らない表現力に協和し過去に例のない新たな世界へと広がり始めている。音楽は、絵や言葉といった他の表現手段と比べ、専門的な要素が多いため別な意味での難しさがあり、表現する人の複雑な経験から生まれるものであるから、その構造は非常に複雑なものであるといえる。

本論文では、数ある過去の音楽等から和音進行をサンプルデータとして用い、それを元に音楽の経験情報をネットワーク状に蓄積する。そして、その経験情報を遺伝的アルゴリズム (Genetic Algorithm: GA) の評価選択部分に用いて、自動的に作曲するシステムを構築しようと試みている。本システムを大きく分けると、経験をデータとして蓄積しておく「経験データ蓄積システム」、和音進行を作成する「コード進行作成システム」、そしてメロディを作成する「メロディ作成システム」の三つから構成されている。

Automatic Composition by Genetic Algorithm

Takushi Yamada and Hisao Shiizuka

Department of Electronic and Information Engineering

Kogakuin University

shiizuka@sin.cc.kogakuin.ac.jp

This paper is attempting at an automatic composition system by the genetic algorithm(GA). Our system consists of three parts:(1)experience data storage system, (2)system for making code advance and (3)system for making melody.

1. まえがき

作曲手順の一つにコード進行を作成し、それにメロディを付ける方法がある。本論文では、この手順に沿ってコード進行作成システムとメロディ作成システムの二つに分けて遺伝的アルゴリズム (以後 GA と記す) を用いて自動作曲システムについて述べている。コード進行作成システムは音楽データをサンプル曲とし、これを多数蓄積したデータ群からコード進行の規則を割り出し、新たなものを生成しようというシステムである。また、メロディ作成システムは、コード進行作成システムで作られたコード進行に、音楽理論に基づいてメロディを付けるシステムである。これらのシステムは、コンピュータで自動的に解を求めることが目的である。ここでは、GA を使用することでデータを進化させ、最良の解 (good solution) を得る方法が示されている。

2. GA と作曲

曲を作るとき、知識や経験といった要素が非常に重要になる。知識や経験が不足していれば、おそらくよい曲は作れないだろう。これを機械にやらせようとしたとき、システムとして「過去のデータ (経験) から適当なものを選び出す

機能」が必要とされる。遺伝的アルゴリズムには「評価」「選択」といった良いデータを得るために重要な思考がある。この評価方法や選択方法が正しくなければ、世代をいくら重ねても良いデータを得ることはできない。このことは、曲を作るときに必要な「経験」や「知識」が不十分の場合によく似ている。この共通点を「GA」と「作曲」の接点として、システム構築を行う。

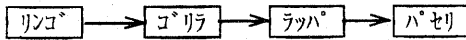
3. 経験ネットワーク

経験ネットワークとは知識のつながりをネットワーク状に表したもので、ある情報から連想される情報をその関係を示す線で結んだものである。ある一つの情報から別の情報を連想するには何らかの要因が存在している。この要因は二つの情報間に一つとは限らず複数存在しているので、ネットワークは各要因のネットを重ねた重複構造になる。この要因をイメージとして考えると、あるイメージに関係する情報がこの経験ネットワークから導出できる。さらに、一つの情報を指定することでイメージにそった次の情報を求めることができ、異なる情報間の関係の深さをネットワーク上の距離で調べることができる。また、情報間のつながりに方向性を持たせることで、情報の推移のある関係で結ぶネットワー

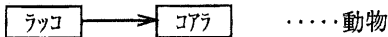
クができる。これは連続的な情報を関係付けたときに有用である。

例えば、「しりとり」の場合、各情報を「言葉」とし、イメージを「言葉の種類（食物や動物など）」として考えると、この経験ネットワークに当てはまる。まず情報として連続する言葉を多数用意し、各言葉の間を有向枝で結びネットワークを作り上げる。そしてネットワークの要素（言葉）と制限（言葉の種類）を指定し、条件に合った言葉を見つけ出す。以下に具体的な例を示す。

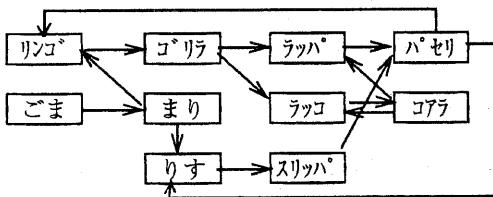
①連続する言葉を用意する



②各言葉の間に関係を持たせる



③ネットワークを作る



「しりとり」の制限を動物や食べ物などとしてもこの経験ネットワークから判断し情報を得ることができる。

4. コード進行作成への適応

コードにはつながりがあり、そのつながりによって曲のイメージが決まってくる。情報をコードに置き換え、イメージをコードの流れから生まれる印象とすれば、この経験ネットワークを使ってコード間の関係を表現することが可能となる。実際にこの経験ネットワークからコード進行を求める場合、次の手順で行う。

- (1) ネットワークを構成するコードから先頭となるもの一つ選ぶ。
- (2) 連結するコードの数を指定する。
- (3) 希望するコード進行のイメージを経験ネットワーク内で使用されているイメージから選ぶ。
- (4) 上の(1)のコードから(2)のコード数に達するまで辿って行く。

4.1 データ化

経験ネットワークをコンピュータで処理する場合、これらをデータ化する必要がある。データ化の方法によって発展性をなくしてしまったり、データ量が必要以上に多くて検索の際に時間がかかってしまう事は望ましくない。特に検索に時間がかかってしまうと、GAの思考時間

に多大な影響を及ぼすため、できる限り短縮する必要がある。ネットワークを表現するのに必要とされる最低限のデータ数に絞り込み、以上のような問題が生じないようにしなければならない。

経験ネットワークの最小単位は、二つの情報とそれを結ぶ1本の線である。したがって、必要なデータは、“情報元”と“情報先”およびこれら二つの情報を結ぶ“有向枝”の三つである。

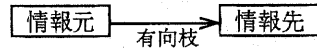


図1 経験ネットワークの最小単位

情報をコード、有向枝をコード進行のイメージ（明るい、暗い等）として経験ネットワークを作る。コード情報には次の種類がある。①コードネーム、②コード構成音、③コードの所属スケール、④コードの根音、⑤コード位置。

コード位置とは、小節単位で数える位置で、曲の先頭や区切り等に使えるコードを見分けるために必要な情報である。これはトニック、サブドミナント、ドミナントを簡単に区別するためのものである。曲には始まりと終わりがあり、途中から曲を聴くと区切りの良いところでない限り、それが曲の途中であることは容易に判断できる。これはコードがもつ機能によるもので、曲が落ち着くコード（トニックコード）と、落ち着かせようとするコード（ドミナントコード）が存在するからである。

コードの区別は、コードネーム、スケール、根音で行う。同じコードでも、スケールが違えば別のコードとして扱う。また、経験ネットワーク内に小節位置以外が全て同じ機能のコードが存在するとき、コード位置情報をそのネットワーク内のコードに付け加える。

5. コード進行作成システム

コード進行は、各コード間のつながりと小節位置により、そのコード進行が正しいか否かが決まる。また、コード進行はそれだけである程度のイメージ（曲の明暗等）を持つと考えられるため、イメージで分類することができる。この関係を経験ネットワークで表現し、GAの遺伝子作成時に参照すれば、出来上がる遺伝子は音楽的に正しいものとなるから、希望イメージに近づくよう評価してGAの思考を繰り返せばよい（図2参照）。

5.1 遺伝子表現(1)

システムの遺伝子にはコード進行をそのまま用いる。コード進行は各コード間に音楽的な規則が存在しているため、遺伝子の各要素は別の要素から影響を受けていることになる。このた

め、各要素単位で突然変異を行ったり、勝手な位置で交叉すると、悪い遺伝子になる可能性が高くなる。これらの点に注意して GA の各処理を行なう必要がある。

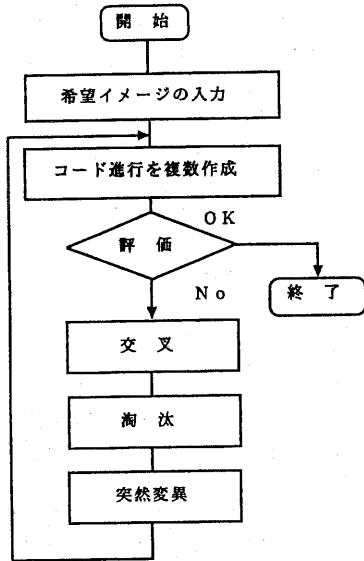


図2 コード進行作成システムの流れ図

遺伝子の作成方法は、経験ネットワークからデータを抽出して行う。経験ネットワークの情報要素から適当に一つ選び出し、その情報から有向枝で結ばれた別の情報へと順に抽出していく(図3参照)。

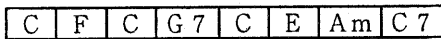


図3 コード遺伝子の例

本システムでは、この情報がコードに対応しているため、以上の作業で得られた遺伝子は音楽的に連結されたコード進行となり、そのつながりは経験ネットワークによって保証されている。したがって、経験ネットワークの作成は慎重に行う必要がある。

5.2 評価方法

遺伝子の評価方法は、本システムを実行する際に指定する“希望する曲のイメージ”に近いものほど高い評価を与える場合と、遺伝子の要素である“各コードの存在位置関係”が正しいか否かによる場合の二つから成っている。希望する曲のイメージは、経験ネットワーク作成時に使ったイメージから選んで、複数イメージの選択、および各イメージの割合を指定することが出来るようにする。仮に、遺伝子の長さを8としたとき、コード進行のコード数は同じ8となる。各コード間にイメージが存在しているた

め、評価対象のイメージ数は七つとなる。

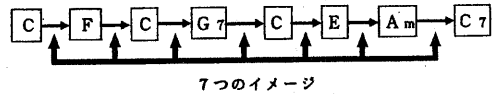
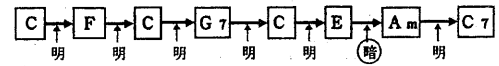


図4 遺伝子の構造

経験ネットワーク内のイメージに“明るい”、“暗い”の二つがあるとき希望する曲のイメージとして、“明るい”を100%、“暗い”を0%と指定したら、評価対象となる七つのイメージが全て“明るい”の遺伝子が最高評価となる。



明...6 暗...1

“明”の評価 = $100 \times 6 \div 7 = 85.7[\%]$

図5 遺伝子の評価例1

“明るい”を50%、“暗い”を50%としたとき、評価対象イメージ数が7であるため割り切れず100%にはなり得ない。この場合は、“明るい”イメージが三つ“暗い”イメージが四つのとき及びその逆のときが最も評価が高くなると考えられる。評価値の最高を100とすれば、この場合の最高値は約85.7%となり、100との差はかなり大きい値となる(図5)。このため、本システムではGA処理の終了条件を“イメージの100%適合”にすると終了できなくなる可能性を含んでいることが分かる。

複数のイメージを希望した場合、遺伝子の評価値は各イメージの中で最も存在率が希望のものから遠いものを対象とする。



イメージ 明...4 暗...1 A...2

【希望イメージ】明 = 50% 暗 = 25% A = 25%

“明”の評価 = $100 \times 4/7 = 57.1 \rightarrow 57.1 - 50 = 7.1$

“暗”の評価 = $100 \times 1/7 = 14.3 \rightarrow 25 - 14.3 = 10.7$

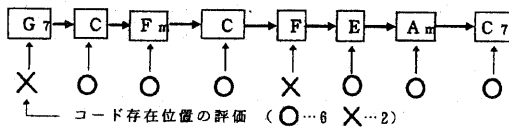
“A”の評価 = $100 \times 2/7 = 28.6 \rightarrow 28.6 - 25 = 3.6$

遺伝子の評価値 = $100 - 10.7 = 89.3$

図6 遺伝子の評価例2

また、遺伝子の要素である各コードの存在位置関係は、経験ネットワーク作成時にコード情報として一緒に記憶させておくもので、コードの機能をコードの位置関係から導き出される。これにより、例えば、曲の頭に使うことができるコード等の区別が可能になる。このコードの位置情報は重要で、これが曖昧では曲のバランスが崩れてしまう可能性がある。このため、コード位置の評価が100%正しい遺伝子であるこ

とが必須となり、前述の曖昧なイメージ評価より優先させる必要がある。すなわち、GA の評価は前述のイメージ存在率で行い、その最高評価遺伝子のコード位置評価が 100 % 正しくなければ、その遺伝子は却下されるようにすればよい。しかし、評価をイメージ存在率だけに固定すると、コード位置の評価は完全に GA から独立し、ランダムに働く要素となってしまう。これを防ぐために、本システムでは、イメージの評価 50 %、コード存在位置の評価 50 % の二つで 100 % とし、コード存在位置の評価も GA の影響が及ぶようにする。



コード存在位置評価 = $100 \times 6/8 = 75$
 コードイメージ評価 = 89.3 (仮の値)
 遺伝子の評価値 = $75/2 + 89.3/2 = 82.2\%$

図7 コード位置関係の評価

しかし、図7の例ではコード存在位置が 100 % ではないので、この遺伝子は本システムから抽出できない。

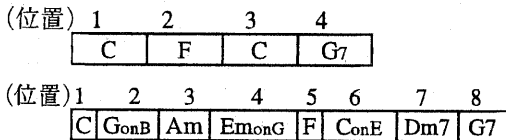


図8 コード位置の例

コード存在位置は 1 ~ 8 で表され、これは小節単位である。具体例及び評価条件を図8および表2に示す。

表2 コード位置条件一覧表

小節	条件を満たすコード存在位置
1	「1」を含み「5」を含まないコード
2	「4」と「8」を含まないコード
3	
4	「4」か「8」を含むコード
5	「1」か「5」を含むコード
6	「4」と「8」を含まないコード
7	
8	「4」か「8」を含むコード

本システムでは、1小節1コードとしてコード進行及びメロディを作成するため、小節単位で機能分けをしている。一般的に、トニックコードは最初の小節で使われる。ドミナントコードはトニックコードの一つ前に現れ、より強い安定感を作り出している。曲の構成を4小節単位でみると、1小節目がトニックコードで、4小節目がドミナントコードになると安定する。勿論、

これ以外の構成もあるが、最も基本的な構成と考え、ここでは、このパターンを中心に進める。また、位置を表すのに1~8まで用いたのは、図8のコード進行のように、5小節目にサブドミナントである「F」が入る場合があるからである。4小節単位では、このコードは小節1であると判断される。これを区別するために1~8とした。

本システムでは一つのコード進行に同じコードが多数出現しないように、ある数を越えると遺伝子自体の評価を最低にしている。これは、二つの情報間が相互に進行可能で、かつコードの位置条件を満たすものが経験ネットワーク内に含まれる場合に起こる。

5.3 交叉方法

遺伝子がコード進行であるため、コード間の関係がある以上、遺伝子要素の横のつながりを無視することができない。このため、交叉位置をランダムに決めるとコード間の関係が保証されなくなり、音楽的に正しいコード進行が得られなくなる可能性が生じる。

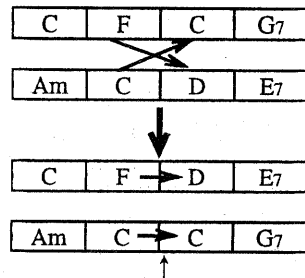


図9 交叉の悪い例

これを防ぐためには、遺伝子要素の位置とその要素内容が同じ所を交叉点とすればよい。すなわち、二つの遺伝子を比べ同じ位置で、かつ同じコードの点を交叉点に取れば、コード間のつながりを保ったまま新しい二つの遺伝子を作ることができる。また、ここでは、複数作られた遺伝子の中で最も評価の高かったものをマスターとし、残りの遺伝子全てと交叉させ、高い評価の遺伝子を残している。

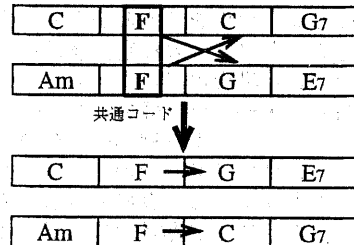


図10 コード遺伝子の交叉方法

5.4 淘汰方法

本システムは経験ネットワークから遺伝子を抽出しているため、経験ネットワークが小さい等の不十分な点があると、作成される遺伝子の種類が少なくなる恐れがある。このような場合、同一処理中に全く同じ遺伝子が出てくる可能性がある。このため、同じ遺伝子同士の交叉が行われ、処理中に存在する遺伝子が同じものばかりになってしまう可能性がある。これを避けるため、同一遺伝子は一つだけ残し、残りを全て削除するようにする。削除されて少なくなった遺伝子は、新たに補充すればよい。また、評価の一番低い遺伝子も淘汰する。

5.5 突然変異の方法

突然変異を行わないと遺伝子が良い方にだけ進化し、その進化方向が間違っていたとき、いくら世代を繰り返しても良いデータは得られない。コード進行作成システムは遺伝子となるコード進行に制約が多いため、一方向に進化してしまう可能性が高い。また、遺伝子の各要素の内容を突然変異で全く別のコードに変化させると、コード間のつながりが音楽的に保証できなくなるため、そのような突然変異は避けなくてはならない。本システムでは、最高評価の遺伝子を小さい確率で削除することで、突然変異処理を行う。確率が大きいと進化の妨げになるので、この確率は小さくする必要がある。

6. メロディ作成システム

メロディ作成システムとは、コード進行作成システムで作られたコード進行に、GA を用いてメロディを付加するシステムである。あるコード進行にメロディを付ける場合、各コードとその進行上に成り立つスケール（音階）を知る必要がある。また実際のメロディは、同じ起伏パターンを小節単位でくり返し使い、聞きやすく耳に残りやすいメロディラインを実現させている。これらはメロディ作りに必要不可欠な要素であり、機械で処理させる場合には、これらをどのように扱うかが重要となってくる。本システムでは、スケール情報をコード毎に経験ネットワークに記憶させ、またメロディの起伏情報はあらかじめ用意している。

GA 処理は1コード単位で行う。コード進行は、複数のコードを連結して先頭のコードから順にメロディを付けていく形となる。

6.1 遺伝子表現(2)

遺伝子には1小節分のメロディを使用する。1小節を16等分し、等分されたメロディ要素をそのまま遺伝子要素とし、遺伝子長16として使用する。本システムでは、1小節の最小単位を16

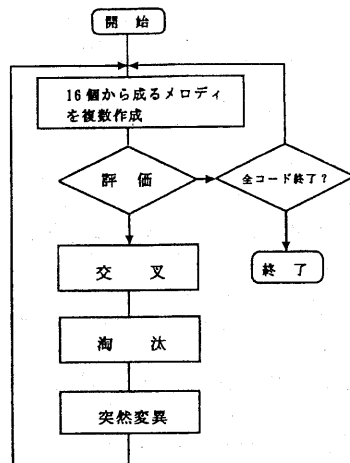


図10 メロディ作成システムの流れ図
分音符に定めて行い、遺伝子の1要素が16分音符に対応している。

0 2 4 2 0 0 2 0 11 9 9 7 5 4 5 7

図11 メロディ遺伝子の表現

表3 メロディ番号対応表

0	ド	[C]	6	ファ#	[F#]
1	ド#	[C#]	7	ソ	[G]
2	レ	[D#]	8	ソ#	[G#]
3	レ#	[D#]	9	ラ	[A]
4	ミ	[E]	10	ラ#	[A#]
5	ファ	[F]	11	シ	[B]

表3は遺伝子要素の数値に対応する音を示した対応表で、全音階の12音で構成されている。遺伝子の各要素は、音程を意味し、0~11の何れかの値が入る。また、図11の遺伝子には向きがあり、左端が小節の先頭音とすれば、その一つ右の要素は16分音符分後の音となる。

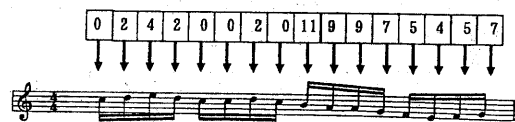


図12 楽譜と遺伝子の比較図

遺伝子の作成方法は遺伝子長16の遺伝子の左端を小節の先頭とし、左端から順に遺伝子要素を決定していく。まず、左端に対応コードが所属するスケールの構成音からランダムに選択する。次に、右隣の要素の決定は、左隣の音とスケール上で隣接した音、もしくは同じ音からランダムに選択する。これは、16分音符の短い間隔で音程が飛躍し、メロディにばらつきが出ないようにするため行うものである。

スケール： 0 2 4 5 7 9 11 (ドレミファソラシの音階)

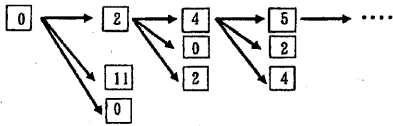


図13 遺伝子要素決定手順

6.2 評価方法

遺伝子の評価は、対応するコードとメロディ音との間にハーモニー的な問題があるか否かで決める。以下に評価条件を示す。

- ①非和声音(コード構成外の音)が連続するのを避ける。
- ②前小節の最終音が今の小節の初音に隣接していると、メロディラインが滑らかなになり良い。
- ③今の小節の最終音が次の小節のコード構成音でない方がよい。
- ④同音の連続を避ける。
- ⑤非和声音は、音階上で隣接する音のつながりの中で使える。即ち、滑らかなメロディライン上に出現させる方がよい。
- ⑥コードに対し短9度(ドとド#等)の音は、音程が連続しているメロディ上でなら使える。
- ⑦今の小節の最終音が短9度になるのを避ける(但し、前音が隣接一つ下音ならOK)。
- ⑧前小節の終わり2音が今の小節の初音を半音で挟む場合は良くない。

前述したように、遺伝子の各要素が16分音符に対応しているので、メロディとしてこの遺伝子を評価する場合、16ヶ所全てを評価対象にする必要はない。メロディに使われる部分だけを評価対象にすればよい。メロディは音を長くのばしたり、細かく別の音に変化させたりして形作られている。メロディ音が別の音に変化するメロディ起伏点を各小節ごとに用意し、その起伏点の遺伝子要素だけを評価すればよい。本システムでは、この起伏点の位置によるイメージの変化については追求していないため、あらかじめ起伏点を用意している。

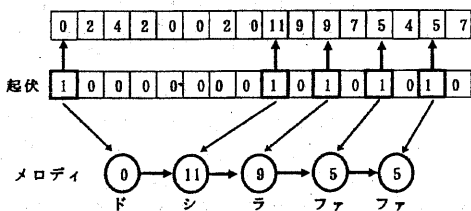


図14 起伏データと評価対象要素

6.3 交叉方法(2)

遺伝子の要素は、16分音符1個分で横につながりがあるため、交叉位置をランダムで決めると、遺伝子作成時に決めた隣接音の音程バラ

スが崩れる。これを防ぐため、本システムはコード進行作成システムと同様に、遺伝子の要素位置が同じで、かつ要素内容が同じ点を交叉点とする。交叉する遺伝子も、最も評価の高かった遺伝子をマスターとして、残りの遺伝子と交叉させる。生成された遺伝子のうち評価の高いものを残していく。

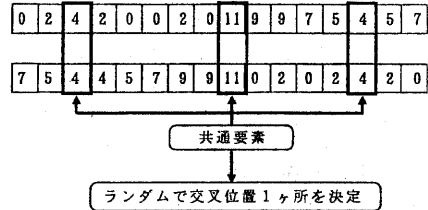


図15 メロディ遺伝子の交叉

6.4 淘汰方法

評価の一番低いものを淘汰する。減った分の遺伝子は新たに補充する。

6.5 突然変異方法

遺伝子は要素間につながりをもたせているため、要素をランダムに変化させると、その関係を崩してしまう場合がある。そのためここでは遺伝子単位ではなくGAの処理単位で突然変異を行う。本システムは、1コード毎に最適メロディ遺伝子を付けているので、小さい確率で一つ前のコードに処理を戻すようにする。評価条件に、コード間のメロディ関係についての評価が含まれているため、あるコードの位置で最適となる遺伝子が作れない状態に陥ることがある。これを防ぐためには、一つ前のコードに処理を戻し、再びそこから作り直すといった処理が必要となり、それを突然変異という形で組み込む。

7. 実行例とその結果

実験ではデータ量を少量で済ませるために、経験ネットワークのデータは便宜上都合の良いデータのみを使っている。これによって、経験ネットワークの構築過程をシステム実行と結果から見る事ができる。

7.1 経験ネットワークの作成

多数のコード進行を用いて経験ネットワークを作成する。経験ネットワークの情報として次の四つを記憶させる：①コード構成音、②コード所属音階、③根音、④コード位置情報。

情報間を結ぶ有向枝は、コードの進行により生まれるイメージ(明るい・暗い等)とする。但し、ここではコード進行とイメージの対応を追求しておらず、“明るい”“暗い”以外のイメージには触れていない。そこで、明らかに“明るい”“暗い”とイメージの異なるコード進行デ

ータについては、その種類分けのみを行いイメージは付けなくておく。コード作成時のイメージ選択は“明るい”“暗い”と、どのコードパターンをイメージを付けなくて組み込むかという形で行う。

7.2 コード進行作成システムの実行と結果

具体的なデータを用いて経験ネットワークを構築し、コード進行作成システムでコード進行を求めていく。

コード進行作成 1: 経験ネットワークへの構成データとして以下のサンプルコード進行を入力する。

イメージ：“明るい”

コード位置	1	2	3	4	1
	C	F	C	G7	C
	C	G	C	G7	C
	C	G	F	G7	C
	C	F	Dm	G7	C

イメージ：“暗い”

コード位置	1	2	3	4	1
	Am	Dm	Am	E7	Am
	Am	Em	Am	E7	Am
	Am	Em	Dm	E7	Am
	Am	Dm	G	E7	Am
	Am	Em	F	E7	Am

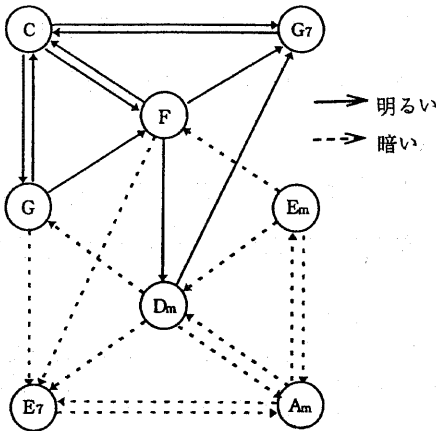


図 16 得られた経験ネットワーク

【実行結果】

実験 1 明るい：50% (2)暗い：50%

Am	Em	F	G7	C	G	F	E7
(2)	(2)	(1)	(1)	(1)	(1)	(1)	(2)

Am	Em	F	G7	C	F	Dm	E7
(2)	(2)	(1)	(1)	(1)	(1)	(1)	(2)

Am	Em	Dm	G7	C	G	F	E7
(2)	(2)	(1)	(1)	(1)	(1)	(1)	(2)

Am	Em	Dm	G7	C	F	Dm	E7
(2)	(2)	(1)	(1)	(1)	(1)	(1)	(2)

C	G	F	E7	Am	Em	F	G7
(1)	(1)	(2)	(2)	(2)	(2)	(1)	

C	G	F	E7	Am	Em	Dm	G7
(1)	(1)	(2)	(2)	(2)	(2)	(1)	

C	F	Dm	E7	Am	Em	F	G7
(1)	(1)	(2)	(2)	(2)	(2)	(1)	

C	F	Dm	E7	Am	Em	Dm	G7
(1)	(1)	(2)	(2)	(2)	(2)	(1)	

コード進行作成 2: 以下のコード進行を実験 1 の経験ネットワークに追加し、新たなコード進行を得る。

イメージ：“暗い”

コード位置	1	2	3	4	1
		G	Am		
				G	Am

実験 2 (1)明るい：50% (2)暗い：50%

C	F	C	G	Am	Dm	G	E7
(1)	(1)	(1)	(2)	(2)	(2)	(2)	

C	F	C	G	Am	Em	Dm	G
(1)	(1)	(1)	(2)	(2)	(2)	(2)	

C	G	C	G	Am	Dm	Am	E7
(1)	(1)	(1)	(2)	(2)	(2)	(2)	

C	F	Dm	G	Am	Em	Dm	G7
(1)	(1)	(2)	(2)	(2)	(2)	(1)	

Am	Em	F	G7	C	G	Am	E7
(2)	(2)	(1)	(1)	(1)	(2)	(2)	

Am	Em	Dm	G7	C	G	Am	E7
(2)	(2)	(1)	(1)	(1)	(2)	(2)	

追加したコード進行は長調（明るい）を短調（暗い）に移行する代表的なもので、実験 1 の“明るい”“暗い”の両コード間をつなぐものである。これを付け加えるだけで出力されるコード進行の種類が増えた。

コード進行作成 3: 以下のコード進行を更に付け加える。このコード進行は“明るい”“暗い”のどちらとも言い切れないので、“パターン 1”として経験ネットワークに組み込む。

イメージ：“パターン 1”

コード位置	1	2	3	4	1
	C	F	Fm	G7	
	C	Fm	C		
		F	Fm	G7	
	C	Fm	C		

【実行結果】

(1)明るい：50% (2)暗い：0% (3)パターン 1：50%

C F Fm G7 C F Fm G7
(1) (3) (3) (1) (1) (3) (3)

C Fm C G C F Fm G7
(3) (3) (1) (1) (1) (3) (3)

C F Fm G7 C Fm C G7
(1) (3) (3) (1) (3) (3) (1)

(1)明るい: 0% (2)暗い: 50% (3)パターン1: 50%

C Fm C G Am Em F E7
(3) (3) (1) (2) (2) (2) (2)

C Fm C G Am Dm Am E7
(3) (3) (1) (2) (2) (2) (2)

(1)明るい: 33.3% (2)暗い: 33.3% (3)パターン1: 33.3%

Am Em Dm G7 C Fm C G7
(2) (2) (1) (1) (3) (3) (1)

C Fm C G Am Em F G7
(3) (3) (1) (2) (2) (2) (1)

Am Em F G7 C F Fm G7
(2) (2) (1) (1) (1) (3) (3)

C F Fm G7 C G Am E7
(1) (3) (3) (1) (1) (2) (2)

コード進行作成 4: 以下のコード進行を更に加える。イメージ名は“パターン2”として経験ネットワークに組み込む。

イメージ: “パターン2”

コード位置	1	2	3	4	5	6	7	8
	C	GonB	- Am	EmonG	- F	- ConE	- Dm	- G7

(1) 明るい: 25% (2) 暗い: 25% (3) パターン 1: 25% (4) パターン 2: 25%

C G Am EmoG F Fm C G
(1) (2) (4) (4) (3) (3) (1)

C Fm C G Am EmonG F E7
(3) (3) (1) (2) (4) (4) (2)

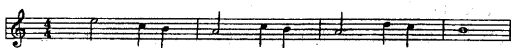
8. メロディ作成システムの実行と結果

前章で得られたコード進行に、メロディ作成システムでメロディを付けた結果を示す。使用するコード進行は、7.で作られた経験ネットワークに、さらにコード進行パターンを数個加えたものから抽出する。紙数の関係でそれらは省略する。

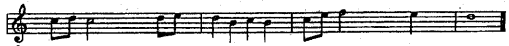
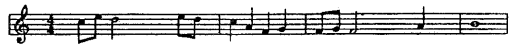
メロディ作成 1: 希望イメージ: 明るい 100%

取得コード: C-F-Dm-G7-C-G-C-G

【結果 1】



【結果 2】



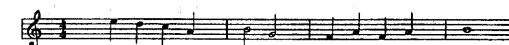
メロディ作成 2: 希望イメージ: 暗い 100%

取得コード: Am-Em-F-E7-Am-Em-Dm-E7

【結果 1】



【結果 2】

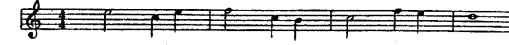


メロディ作成 3: 希望イメージ: 明るい 50%、

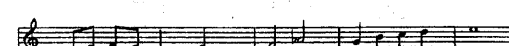
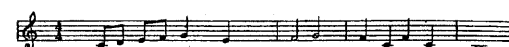
暗い 50%

取得コード: C-F-Fm-G7-C-Fm-G7-C

【結果 1】



【結果 2】



9. むすび

本論文で提案した GA による自動作曲の骨子は、①経験ネットワーク、②コード進行作成システム、③メロディ作成システムの三つである。

これらの内容は今後改善の余地が残されているが、特に、イメージ通りの曲を出力させるためには、以下の点も重要な課題であると思われる。

- (1) リズムとイメージの関係
- (2) テンポとイメージの関係
- (3) 音色とイメージの関係

本論文で扱ったのは、コード進行により生まれるイメージのみである。また、各コードのイメージや起伏点とイメージの関係、評価条件の種類とイメージの関係も重要で、これらのイメージの関係を全て定めたとき、初めて希望通りの曲が得られるシステムになるであろう。それらは今後の課題としたい。