2006－MUS－68 (6)
2006／12／15

ECを用いた作曲支援システムと作曲モデルの客観的評価手法

安藤 大地　丹治信　伊庭斉志

東京大学大学院新領域創成科学研究科基盤情報学専攻

今までIECを用いて作曲支援システムを構築しようとするとき，どのような作曲モデルがターゲットとしているユーザに適しているのかの明確な基準はなかった．そこで，著者らはECを用いて作曲モデルの客観的な評価を行う基準を得る手法の提案を行う．また，IEC作曲支援システムのユーザインタフェースの改良を行い，より効率的に作曲が行えるようになった．

Generation and Objective Evaluation of Composition Model by means of EC

Daichi Ando　Makoto Tanji　Hitoshi Iba†

Dept. of Frontier Informatics, Graduate School of Frontier Sciences,
The University of Tokyo.

To solve problem that there are no clear criteria in deciding whether a composition model is suitable for IEC based composition system, the authors propose a technique which can be evaluate composition models objectively. Also the authors present new user-interface for GP based IEC compostion system.

1　導入

本稿では，対話型進化論的計算 (Interactive Evolutionary Computation, IEC) を用いた作曲支援システム CACIE(Computer-Aided Composition by means of Interactive Evolution)[10] の，主にユーザインタフェースについての改良と，作曲支援システムの客観的な評価手法の提案を行う．

1.1　対話型進化論的計算

対話型進化論的計算とは，遺伝的アルゴリズム (Genetic Algorithm, GA) や遺伝的プログラミング (Genetic Programming, GP) などの進化論的計算 (Evolutionary Computation, EC) を対話型にしたものである．

通常のECにおいては，各個体の評価値，すなわち環境への適合度は評価関数によって自動的に与えられる．評価関数の設計自体が，ECを用いて問題を解く手法であるとも言える．

ところが，芸術作品の創作などにECを用いた場合，作者が事前に評価関数を設計しなければならない．この場合の評価関数は，作者が求めている作品そのものにどれだけ近いかを表したものになる．しかしながら，最初から求めている作品が決まっているのであれば，ECを用いて作品創作を行う必要はない．芸術作品の創作に偶然性を取り込むメリットは，人間の創作者が思いもつかなかった作品を手に入れられることにある．作品の完成形が最初から決まっているのであれば，偶然性の入り込む余地はない．しかし，ECは評価関数が決まっていないと走らせることが不可能であるので，偶然性を期待して芸術作品の創作にECを用いたいのに用いることができないという問題が生じた．

そこで，人間の創作者本人が評価関数の役目を果たし，各個体に適合値を与える事により，ECによって芸術作品創作を行う手法が開発された．Dawkinsは採用するかしないかの二値評価を用いて，生物のようなコンピュータグラフィックを生成するプログラムをECによって合成する研究を行った [5, 6]．この，人間が評価を与える方法は，システムと人間の対話によって成り立っているため，この手法を対話型進化論的計算と呼ぶ．

Dawkinsのシステムは二値評価を用いた模擬育種法 (Simulated Breeding) であったが，段階的な評価値を

用いる手法もその後多く開発されている．また評価の与え方についても，段階が多すぎるとユーザに負担を与えるなどの結果が報告されている [8]．

1.2 IEC の音楽創作への応用

しかし，音楽創作に IEC を応用しようとした場合，Dawkins のようなコンピュータグラフィックの合成では問題にならなかった問題が，いくつか生じる．

まず，評価に要する時間が膨大になることがある．グラフィックなどのメディアの場合，ユーザは個体の表現形を一目見て評価値を大体決める事が可能である．しかしながら，音楽の創作の場合，基本的に最後まで生成された楽曲を試聴する必要がある．そのため評価に時間がかかることになり，ユーザ負担の増大を招く．

これに対して，最初に音楽創作に EC を適用した Biles の GenJam システム [1, 2] では，個体に評価を与えるインターフェイスを教師と生徒の対話のような形で実装し，途中で演奏を打ち切らせることを可能にし，ユーザ負担の軽減を狙った．

また音楽生成特有の問題として，音楽という構造的なものを対象として扱う以上，構造的な遺伝子型と遺伝子操作を用いる必要があるということがある．単純な GA では収束に時間がかかり，実用にはならない．また楽曲が大きなものになると，それだけ染色体が肥大化してしまい，収束に要する時間だけでなく，発現させる処理にかかる時間や，評価に要する時間も多く必要となる．このような時間のロスは対話型のシステムに取っては，大きな問題となる．

Dahlsted は再起的に展開していく木構造の遺伝子型を用いる事で，シンプルな染色体で複雑な旋律を表現する事を可能にした [4]．

その他にも，対話型であることの弱点として，ユーザの求めるものに近いものから求めるものまでに進化するために必要な時間がかかることがある．抽象画のようなコンピュータグラフィックでは染色体上の小さな違いはさほど問題にはならないが，音楽創作，特に構造的な遺伝子型を用いた場合は，音楽的に大きな違いを生む事が多い．

1.3 作曲支援システム CACIE

前述のように，以前筆者らは IEC を用いた作曲支援システム CACIE の開発についての報告を行った [10]．

CACIE では，1.2 項で述べたような IEC を音楽創作に応用する際に問題となるいくつかの点を改善したものである．CACIE は，音楽的に意味のあるツリー構造遺伝子型と遺伝子操作，マルチフィールドユーザインタフェースによるユーザの進化への積極的な介入，などの特徴を持っている．

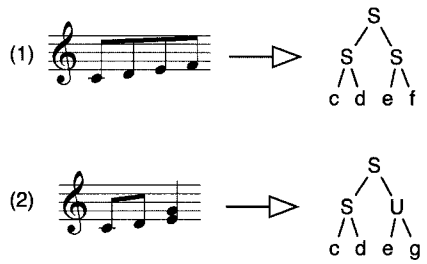

図 1: Tree Representation of Musical Tree.

1.3.1 ツリー構造遺伝子型と遺伝子操作

図 1 の様なツリー構造遺伝子型を採用している．終端ノードはそれぞれ音符か音列を表現し，非終端ノード (関数ノード) は自分の子ノードの音列をどのように接続するかを表す．図 1 における非終端ノード S は二つの音列を続けて演奏することを表し，ノード U は二つの音列を同時に演奏することを表す．非終端ノードは音楽的に意味のある様々なものを用意しており，またライブラリとしてもユーザに提供している．ユーザは既存の非終端ノードを組み合わせて新しい接続方法を取り込む事が可能になっている．

また，特殊な終端ノードである，再起終端ノードとよばれる仕組みを開発した．再起終端ノードは表現型への変換の直前に自分を自分の親ノードをルートとするサブツリーで置き換える．図 2 の `R_e` が再起終端ノードである．この仕組みにより，シンプルな染色体で音楽的な繰り返しを表現することが可能になった．

同様に通常の GP の突然変異に加え，音楽的に意味のある突然変異として，Increase と Decrease と呼ばれる突然変異を導入している．Increase は再起終端ノードの突然変異としての実装であり，Decrease はその反対の働きをする．

1.3.2 マルチフィールドユーザインタフェース

CACIE では，生成された子供世代が親世代をすぐに置き換える事がないマルチフィールドユーザインタフェース [9] を採用している．さらに Genome Storage と呼ばれる染色体の一時保管庫を設け，気に入った染色体を別の系統の進化の流れに挿入することなどを可能にしている．また，作曲過程の複合構造を採用することで，通常のクラシック音楽の作曲過程に近づけるともに，染色体の肥大化によるユーザ負担の増大や処理速度の低下を防いでいる．

これらの特徴によって，CACIE ではユーザがより強い干渉を進化に対して与える事が可能になり，偶然

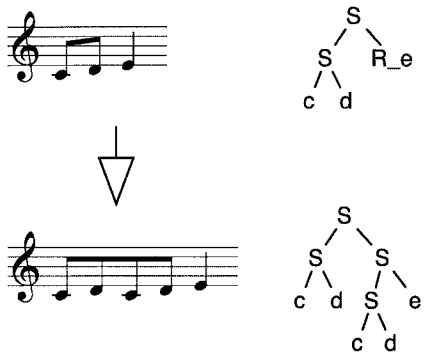

図 2: Development of Recursive Terminal Node.

性を取り入れながらも，作曲者の思い通りの楽曲を合成することができる．

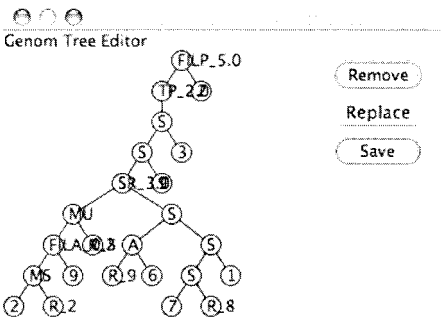

図 3: Tree Presentor and Editor.

2　ユーザインタフェースの改良

本項では，今回行った CACIE へのユーザインタフェースの改良を説明する．

2.1　染色体ツリーの提示

対話型 GP の場合は，生成されたプログラム遺伝子そのものをユーザに提示することにより，聴取をしなくても生成された楽曲の概要がわかることで，ユーザ負担の軽減にも繋がる．

従来の CACIE では，染色体ツリーは S 式の文字列として表示されているだけであった．文字列の状態でも概要を把握することは可能であったが，複雑になると構造の把握が困難になり，自分でツリーを書き起こす必要があった．そこで，新しいユーザインタフェースとして，染色体ツリーを実際にツリー状に表示することを可能にするウィンドウを設けた．これにより，より直感的に染色体の構造，すなわち生成された楽曲の構造を把握することができ，評価の際の負担を軽減することができた．

図 3 にツリー表示ウィンドウを示す．

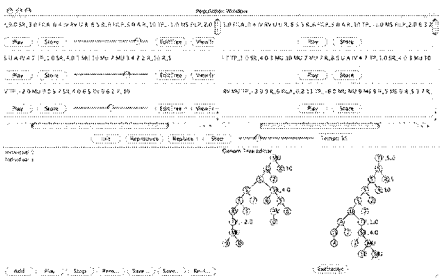

図 4: Main Window of CACIE.

2.2　染色体ツリーの GUI での修正

また，図 3 では，染色体ツリーの手動での変更を行うことができる．Unemi らが提案したグラフィック生

成のためのIECシステムで実装された手動での変更やマスキングなどの機能[9]を，音楽創作にも応用したものである．従来のCACIEでは，一度染色体ツリーをテキストファイルに書き出して，それをテキスト処理により編集し，もう一度CACIEシステムに取り込む必要があった．このインターフェイスの実装により，直接グラフィカルに染色体ツリーを編集することが可能になった．

ECでは，遺伝子操作は決められた範囲内でランダムに適用される．従って，作者が提示された個体に対して微小な変更を望む場合，その微小な変更をされた個体を手に入れることは非常に難しい．そこでこのような手動編集機能を実装した．

2.3 手動による交叉

同様の理由から，手動による交叉も実装した．図4に示したウィンドウの右下部分は，二つの染色体ツリーを同時に表示し，任意のサブツリーを交換することができる交叉を手動で行う事が出来る．

3 作曲モデルとなる遺伝子型の評価

3.1 作曲モデルの評価基準

2項で述べた，ユーザインタフェースについては，実際に使用をしてもらい，アンケートなどを取る事である程度客観的で定量的な評価が可能となる．

しかしながら，CACIEの基礎となる遺伝子型，すなわち作曲モデルは，客観的かつ定量的な評価を行いにくい．

既にECを用いて作曲を行う研究は多々ある[3]．しかしながら，どの遺伝子コーディングがどのような曲の合成に適しているのかの明確な基準は示されてこなかった．また，ECを用いた作曲システムのみならず，音楽創作プログラミング言語のような，その他のテクニックを用いた作曲システム，自動作曲システムについても，同様の問題がある．このような作曲システムや作曲支援システムでは，そもそもシステム自体が作曲家が自分の作品のために構築したものがほとんどであり，その評価は，作曲家自身の極めて主観的なコメントによりなされることが多い．

これらの遺伝子コーディングや音楽創作プログラミング言語は，前述のように作曲モデルと呼ぶことも可能である．

前述のように，これらの作曲モデルの妥当性を客観的に評価できなければ，自分でシステムを開発し自分の楽曲創作へ用いる作曲家以外の人間には，その遺伝子コーディングや自動作曲システムの妥当性はわからない．CACIEにおいても，[10]で示した合成実験，特にピアノの小品の合成では，作曲手順などは主観的な要因によって決められたものが多い．このような評価は，ユーザインタフェースなどの諸機能を含めた評価ならば問題はないが，遺伝子型そのものの評価については不十分であった．

そこで，現在CACIEなどのEC作曲支援システムの評価方法として検討を進めている，客観的な指標を導く基準をECを用いて発見する手法を提案する．

3.2 提案手法

現在実装を進めている提案手法は，作曲モデルをECシステムに実装し，ターゲットとなる曲を合成し，答えにたどり着くまでに要した世代数などの結果を見る，というものである．EC自体は，どのような形の作曲モデルでも遺伝子として取り込むことができるため，種々の作曲モデルを検証する手法として，容易に用いることができる．

実際に人間の作曲家が作曲作業として行う，複数のパターンの生成を行い，評価して，いいものを採用するという行動モデルは，ECのシナリオと同じであり，人間の作曲行動は，IECを用いた作曲支援システムのように，そのままEC上に実装することが可能である．

音楽創作プログラミング言語などはGPの形で実装し，乱数テーブルを用いた作曲法は，GAで実装することが望ましい．

各作曲モデルの実装対象となるECは，作曲モデル以外の違いを少なくするため，全く同じフレームワークを用いる必要がある．また，ECの遺伝子操作はランダムに適用されるため，数回の試行の平均を取る必要がある．

3.3 実装と実験

3.3.1 フレームワーク

実験のために，GAやGPなどを統一した枠組みで扱うことができ，なおかつ作曲モデルを容易に実装可能にするため，拡張が非常に容易なECフレームワークを設計し，実装を行った．

3.3.2 ターゲット楽曲

現在著者らが評価のターゲットとして考えている楽曲は，J. S. Bachの15のインベンションから8番である．この楽曲は，両手パートとも単旋律，2パートによって構成されており，テーマの変奏や展開，音高の上げ下げを伴ったゼクエンツが，両手パートともにはっきりと確認できるため，音楽的構造がわかりやす

い．なおかつ 34 小節とそれほど長い曲ではなく合成自体が困難であるとは考えられない．そのため，テストケースとしては最適だと判断した．

前述の通り，音高の配列は，最低音を 0 とし，そこからの度数表記で各音高を取得する．最低音はヘ音下第二線の C，最高音はト音上第二線 C である．最高は最低音から数えて完全 27 度上 (28 ステップ分) にあたる．音符総数は 596 個である．この音高の配列から各音符間の音程 (音高差) の配列を導きだし，それを同定する．

3.4 比較対象となる遺伝子型

CACIE と比較を行うことを検討しているモデルについて述べる．

3.4.1 ランダム生成をモチーフとした遺伝子型

まず，比較対象とするべき遺伝子型は，固定長の整数の配列であり各整数 (各遺伝子座) の値がランダムに決まっているものである．これはあらかじめ決められたテーブルなどを一切用いずに，多面体サイコロを振ってその出た目を音高とするという作曲法のモデルである．

この作曲法には，音楽的な知識が一切取り込まれておらず，音楽的な構造を導きだすのは難しいと思われる．しかしながら，音楽構造を遺伝子構造に取り込んだモデルとの比較のためには必要だと思われる．

3.4.2 GP のプログラムによる数式表現の旋律

CACIE の遺伝子型は GP のプログラム生成の概念を基にしている．そのため一般的な GP を用いた手法と比較し優位性を示さなければならない．

Laine の数式表現による楽譜表現の合成は，典型的な関数の合成などに用いられる GP を用いて旋律を合成した試みである [7]．用いられる終端ノードや非終端ノード，ターゲットとなるプログラムは，GP による関数同定と同じものである．

そのため，このモデルは Csound などの Unit Generator 言語のスコアファイルをプログラミング言語を用いてアルゴリズム作曲で生成するのと同じであると考えることが可能である．特に GP であるため，CommonLisp Music でスコアファイルのみを生成するようなケースに非常に似ていると言える．

3.5 提案手法の問題点

問題となるのは，実験における作曲モデル以外の要因が結果に及ぼす影響である．作曲モデルの以外の違いをなるべく生まないように，拡張が容易な EC 全般向けのフレームワークを設計，実装して実験を行ったが，遺伝子コーディングに依存する遺伝子操作などの違いは吸収することができない．

例えば，今回の実験では固定長のバイナリ配列を遺伝子，染色体として GA の手法を用いて実装を行った．そのため GA で交叉法として一般的である一点交叉を用いた．突然変異として用いたビット反転に関しても GA では一般的である．しかしながら音楽創作プログラミング言語などを実験の対象とする場合，GP の手法を用いて実装を行うため，交叉法は部分木の交換などが用いられることになる．GP における突然変異の部分木の逆位などの概念も GA にはない，特殊なものとなってしまう．

このような GP 特有の遺伝子操作は，GP の「構造を持った遺伝子，染色体を扱う」というポリシーからきているもので，構造があることが前提である楽曲合成では当然使用すべきものである．GP でも部分木を全く無視し，一点交叉や，ビット反転に相当するようなラベルの張り替えをを用いる事は可能ではあるが，実験の趣旨にあわない．

また，イントロンの問題もある．イントロンとは染色体の中で，実際に発現型 (ここでは生成された楽曲) には影響を与えない部分のことである．実際の生物の進化や EC では，このイントロンが重要な役割を果たすことがわかっている．一定の期間，突然変異や交叉の変化を発現型に表れない部分に溜め込み，時期が来たら爆発的に進化を加速させる効果があると言われている．

GP では，自然的にイントロンが生成され，これがビルディングブロックなどの構成に関わり，収束効率に大きな影響を与えている．ところが，3.4.1 で示したような，サイコロの出目をそのまま音高と読み替えるような作曲モデルでは，イントロンは発生しない．

これらの違いをどのように吸収し，説明していくかが重要となる．

4 まとめ

本稿では，EC を用いた作曲支援システム CACIE のユーザインタフェースの改良と，現在実装を進めている遺伝子型自体の客観的な評価手法についての報告を行った．

インタフェースの改良については，これからアンケートなどをより多く取り，評価を行っていきたいと考える．また遺伝子型の客観的評価では，多くの比較

対象となる作曲モデルをフレームワークに実装し，実験を行っていく．

参考文献

[1] J. Biles. Genjam: A genetic algorithm for genrating jazz solos. In *Proceedings of 1994 International Computer Music Conference*, Arhus, 1994. ICMA.

[2] J. Biles, P. G. Anderson, and L. W. Loggi. Neural network fitness functions for a musical iga. In *Proceedings of IIA'96/SOCO'96. Int'l ICSC Symposia on Intelligent Industrial Automation And Soft Computing, Reading, UK*, pp. B39–44, 1996.

[3] A. R Burton and T. Vladimirova. Generation of musical sequences with genetic techniques. *Computer Music Journal*, Vol. 24, No. 4, pp. 59–73, 1999.

[4] P. Dahlstedt and M. G. Nordahl. Augumented creativity: Evolution of musical score material, 2004.

[5] R. Dawkins. *The Blind Watchmaker*. Longman, Essex, 1986.

[6] R. Dawkins. The evolution of evolvability. In C. G. Langton, editor, *Artificial Life*, pp. 201–220. Addison-Wesley, 1989.

[7] P. Laine and M Kuuskankare. Genetic algorithms in musical style oriented generation. In *Proceedings of First IEEE Conference on Evolutionary Computation*, pp. 858–861, Washington D.C., 1994. IEEE.

[8] H. Takagi and K. Ohya. Discrete fitness values for improving the human interface in an interactive ga. In *Proceedings of IEEE 3rd International Conference on Evolutionary Computation (ICEC'96)*, pp. 109–112, Nagoya, 1996. IEEE.

[9] T Unemi. A design of multi-field user interface for simulated breeding. In *Proceedings of 3rd Asian Fuzzy System Symposium: The Korea Fuzzy Logic and Intelligent Systems*, 1998.

[10] 安藤大地, P. Dahlstedt, M. G. Nordahl, 伊庭斉志. 対話型 gp を用いたクラシック音楽のための作曲支援システム. 芸術科学会論文誌, Vol. 4, No. 2, pp. 77–86, 2005.