

蓮井洋志[†] 長島知正[‡]室蘭工業大学情報工学科[§]

1 はじめに

遺伝的アルゴリズム (GA) とは、生物の進化のモデルを基本としている問題解決方法である。最適化問題を多項式時間で解くための一手法である。最適化問題には NP 完全な問題がある。NP 完全な問題を解くには、多くの場合、線形探索やランダム探索では多項式時間で解決できない。良いアルゴリズムが発見されていない問題では、規模が大きいときには GA を使わずに解くことは、実現上無理である。

GA は局所最適解を避け、大域解に落ち着くアルゴリズムである。複数の解のある最適化問題を解くには適していない。そこで高速にしかも多数の局所解を探索するアメーバ型 GA を提案する。アメーバ型 GA とはアメーバの進化のモデルを模擬したアルゴリズムのことである。アメーバは性別を持たず、その後に進化した生物を多く持つ。つまり、交叉がなく解が多いと言うことである。また、突然変異率が多く、その後進化して生まれた動物よりは大きい。適応度の最大値と思われる値に近づいた遺伝子を問題の解と見なす。つまり、解は1つだけではなく、多く存在する。

我々はアメーバ型 GA を自動作曲システムに応用した。自動作曲システムは音高の列、つまり音列とリズムの2つの方向からメロディーを特定する。この中の音高の列を生成する音列生成システムでは、遺伝子を音列と見なす。この遺伝子は、塩基が音高の値を表す整数となる。コード構成音や前の音高の差などに点数を与え、その点数の総計を適応度とし、最大適応度の0.9999倍以上である解を答とする。人間の感性に訴えかけるメロディーは一通りでないように、音列生成システムででき上がる音列は多数ある。GA を自動作曲に応用する例 [1] は多く存在するが、アメーバ型 GA を応用したものはない。

本稿では、アメーバ型 GA を提案し、それをどのように自動作曲システムに応用したかを書く。第2章ではアメーバ型 GA について、第3章では自動作曲シ

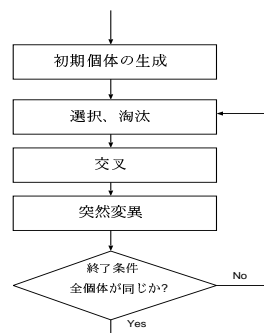


図 1: 基本 GA

ステムへの応用例を説明する。第4章に、アメーバ型 GA の基本 GA やランダム探索との比較評価実験をおこなう。

2 アメーバ型 GA の提案

2.1 基本 GA の問題点

基本 GA は図1で表されるアルゴリズムである。数は問題の規模にもよるが、まず、初期個体を100前後ランダムに生成する。その後、その個体の適応度を計算し、その値の大きさによって個体の選択、淘汰をおこなう。値の大きいものは増えるが、その分値の小さい個体が淘汰される。

次に、2つの個体の遺伝子の一部分を入れ換える。これを交叉と呼ぶ。その後、突然変異で遺伝子の一部がごく稀な確率で変わる。最後に、終了条件として同じ個体の数が大勢をしめた時点でアルゴリズムが終了する。終了条件を充たさない場合は個体の選択、淘汰に処理が戻る。

基本 GA では、問題の解が複数ある場合、1つの個体が大数を占めにくい。つまり、複数解を持つ問題を解くのは難しい。

2.2 アメーバ型 GA

本研究ではアメーバの進化になぞらえた GA を提案する。アメーバには性別が無いために交叉がない。個体をふやすときには細胞分裂する。淘汰の後の突然変異は細胞分裂によって起こる。アメーバにおいては、細

*Propotion of Amoeba-Type GA and Application for Automatic Composing System, Sonneteer with it

[†]Hiroshi Hasui

[‡]Tomomasa Nagasima

[§]Department of Computer Science and Systems Engineering in Muroran Institute of Technology

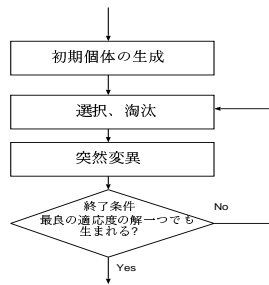


図 2: アメーバ型 GA

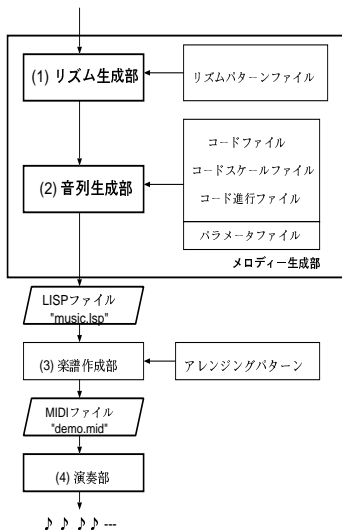


図 3: 自動作曲システム Sonneteer の構成

胞分裂した個体は死なない。全ての個体を淘汰せずに、優秀な個体は次世代に残す。エリート保存戦略である。細胞分裂した後の個体には突然変異を行わない。

アメーバの突然変異率は動物などよりも大きい。そのため、アメーバ型 GA では基本 GA と比較して突然変異率が大きい。

図 2 に、アメーバ型 GA の処理の手順を書いてある。基本 GA とちがう点の一つには交叉が無いこと、エリート保存戦略であること、終了条件が最大適応度に近い閾値以上の個体が個体集団の中で 1 つでも存在すれば解とすることの 3 つである。その結果として、アメーバ型 GA は複数の解がある問題を高速に解くことができる。

3 自動作曲システム Sonneteer

3.1 自動作曲システムの構成

図 3 に自動作曲システム Sonneteer の構成図を書く。

この自動作曲システムは Red Hat Linux 上で開発した。そのまま、Windows-XP の Visual C++ 6.0 でも正常に動作した。

Sonneteer 内でのデータの流れを説明する。メロディー生成部で (1) リズムを決定し、(2) それにあった音列を生成し、最後にその 2 つの要素を融合し、“music.lsp” にそのデータをダウンロードする。その後、(3) そのデータを “demo.mid” に変換する。(4) 演奏部では、それを TiMidity が演奏する。

音列生成部は、アメーバ GA で一回の手続きで 1 フレーズ分の音列を生成する。1 フレーズ毎にコード進行が決まっています、その進行におけるコードを中心にメロディーを作る。遺伝子として音列を定義した。1 つの遺伝子座には 1 つの音高を表す数字が入る。

4 考察

基本 GA では、適応度が 1 である遺伝子で収束せずに、0.98 くらいで収束する場合が多い。つまり、基本 GA では適応度が 1 の解が求まらない。これは誤った解を探索することになる。しかし、アメーバ型 GA ではおおむねその可能性がない。どちらが正しい解を探索しているのか分かりにくい。生物の進化は最良の解を出していないのではないのか。

ランダム探索とアメーバ型 GA は淘汰、選択、突然変異をぬかせば同じような解法である。アメーバ型 GAの方が実行速度が速い。明らかに遺伝的操作が目的解へ収束する力を持っていることを表す。

GA を活用した自動作曲システムの研究は数多い。この方式では、リズムと音列を合わせてメロディーを作り出す。しかし、アメーバ型 GA を使ったものは無い。感動するメロディーは一つだけではなく、複数が存在するものである。そのため、普通の GA では解が一つに収束しないために終了しない。

適応度が最大の遺伝子が正しい解である。終了条件が大方の遺伝子が同じものとなるかどうかであると、収束するのに時間がかかるのみならず、本当に正しい解に辿り着くかどうか保障されない。遺伝子がただの 1 つでも適応度が 1 の解を見つけたときに、収束すべきではないかと考える。

参考文献

- [1] 田中 健, 外山 史, 東海林 健二. 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み合わせによる自動作曲. 情報処理学会音楽情報科学研究会報告, 41-8, pp. 43-48, 2001