

2 Z E - 0 4 遺伝的プログラミングを用いた協調行動の創発

— イントロンの制御と遺伝子移民の有効性 —

寺尾 真 石塚 満 伊庭 斉志

東京大学工学部電子情報工学科

1. はじめに

本研究の目的は複数のエージェントが存在する問題において、遺伝的プログラミング(GP)を用いてエージェント間で協調行動を行うプログラムを獲得することである。GPとは進化論的な考え方に基いて問題に対する最適プログラムを探索する手法である[1]。本論文ではより効率的に優秀なプログラムを獲得する方法として、次の2手法を提案する。

(1) 遺伝子中のイントロンの制御

(2) エージェント間での遺伝子移民

イントロンとは遺伝子の実際には発現しない部分のことである。GPの実行を進めると遺伝子が次第にイントロンで占められていき、GPの探索が事実上停止してしまうことが経験的に知られている[2]。そこで本研究では実行時イントロンという考えを導入し、これを削除することでイントロンの生成を抑えることを試みる。

また、複数のエージェントのプログラムを進化させるときに、あるエージェントのための優秀な遺伝子を他のエージェントのための遺伝子群の中に送り込む(移民させる)ことで、効率的に協調プログラムが得られることを示す。

本研究ではロボットナビゲーション問題を例に、以上の2手法の有効性を実験的に検証する。

2. 実行時イントロン

遺伝子中の明らかなイントロンには、

(×2 (÷2 A))

のような意味的イントロンや、

(IF TRUE A B)

のような構文的イントロンが挙げられる。しかし、このように明示的なイントロンには該当しないが、遺伝子を実際に解釈する段階では実行されない、というイントロンがGP実行の際には大量に生ずる。

現実にはこのような明示的でないイントロンの方が、効率的な探索を大きく阻害していると考えられる。そこで本研究では、このように遺伝子を実際に解釈するときには実行されない部分を実行時イントロンと定義し、これを世代毎に削除することにした。構文的イントロンはそれを全て発見するのが困難であるが、実行時イントロンは遺伝子中のコードに実行が来たかどうかをチェックするだけなので、簡単に識別することができる。なお、実行時イントロンはその定義から構文的イントロンを完全に含んでいるが、意味的イントロンは該当しない。

3. エージェント間での遺伝子移民

本研究ではマルチエージェント問題を扱っているため、複数のエージェントのためのプログラムを独立に進化させている。しかし、各エージェントに要求されている最適な役割が異なっても、共通に必要とされている要素があるならば、その部分はエージェント間で協力して獲得した方が効率が良いと考えられる。例えば、サッカーの選手に要求されている行動はポジションによって異なるが、ドリブルなどの基本的な動作は共通である。そこで本研究では、このドリブルに相当する共通プログラムを協力して獲得するために、各エージェントを制御する遺伝子の中で優秀なものを他のエージェントのための遺伝子の中に移民させる実験を行った。

4. ロボットナビゲーション問題による実験

ロボットナビゲーション問題とは、フィールド上に配置されたロボットがそれぞれ決められた目的地に向かうという問題である。各ロボットはお互いに進路を妨害しないように協力する必要がある。本研究の実験では、図1のようなフィールドを用いて連続値座標上でのシミュレーションを行った。地点

a,b,c,d から出発した 4 体のロボットはそれぞれ地点 b,a,d,c を目指す。実際には全部で 6 つの訓練フィールドを用いてプログラムを進化させた。

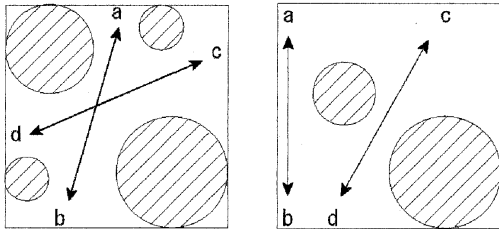


図1 訓練フィールドの例

獲得されたプログラムの適応度には次を与えた。

$$\begin{aligned} \text{適応度} &= 100.0 - \text{残りタイムカウント} \\ & \quad (\text{時間内に全ロボットが目的地に到達した時}) \\ &= 300.0 + \text{残り距離の総和} \\ & \quad (\text{時間内に目的地に到達できなかった時}) \end{aligned}$$

従って適応度は小さいほど良いことになる。

このような実験例を用いて通常の GP を行った場合、イントロンの制御を行った場合、遺伝子移民を行った場合、両方を行った場合の 4 つについてそれぞれ 40 回実験したときの適応度変化の平均グラフが図 2 である。また、時間内に全ロボットが目的地に到達するプログラムまで進化した回数はそれぞれ 5 回、13 回、15 回、26 回であった。これらを見るとイントロンの制御を行った場合、遺伝子移民を行った場合のそれぞれにおいて、通常よりも良い成績をより早い段階であげていることがわかる。

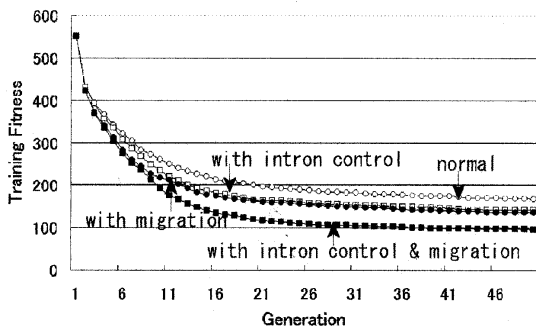


図2 適応度の変化

また、図 3 は各世代で最も優秀な成績を上げた遺伝子のサイズの変化グラフであるが、イントロンを削除している場合はよりコンパクトなプログラムが得られていることがわかる。

図 4 は両手法を用いて得られたプログラムによるロボットの行動の一例である。矢印で示された軌跡は

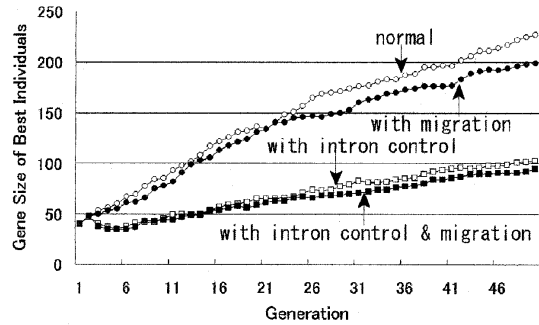


図3 最良遺伝子のサイズの変化

同じロボットの行動である。このロボットは他のロボットが近付いてきても自分は避けず、相手に避けさせている。このようにロボットによって役割分担がなされていることが確認された。さらに、障害物をよけて目的地に移動するという基本動作は各ロボットで似たような行動が見られ、遺伝子移民による基本プログラムの共有が行われていると考えられる。

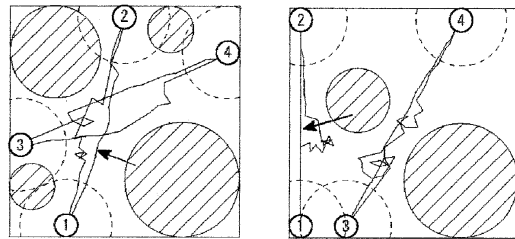


図4 獲得されたプログラムの行動図の例

また、進化によって得られたプログラムを訓練例とは異なるフィールドで動かしてみたが、イントロンの削除や遺伝子移民によってプログラムの頑強性が失われることもなかった。

5. おわりに

本研究ではロボットナビゲーション問題を例に、イントロンの制御と遺伝子移民を行うことによって、より優秀でコンパクトなプログラムを獲得できることを示した。今後、これらの操作が遺伝子レベルでどのような影響を与えているのかを調べることで、より有効な方法に改善していくことが課題である。

参考文献

- [1] 伊庭斉志「遺伝的プログラミング」東京電機大学出版局 1996
- [2] 伊庭斉志「進化論的計算の方法」東京大学出版会 1999