

5 E - 6

Genetic Algorithm における 遺伝子コーディングの最適化に関する研究

杉山 智則

慶應義塾大学 理工学研究科 計算機科学専攻
修士課程 2年 中西研究室

平成4年10月

1 はじめに

自然選択と突然変異による進化論は科学的に正しいと証明されたわけではないがその理論は多くの人に支持されている。この自然選択と突然変異によつての進化に強く影響を受け、それをプログラムに応用しようと生まれてきたのが Genetic Algorithm である。

Genetic Algorithm は表現型と呼ばれる個々の個体を適合度関数によつてそれぞれ評価し、高い評価の表現型の遺伝子を多く残すように次の世代を作り、それを繰り返すことによつて優れた表現型ばかりの世代ができるという方法である。

子として表現するかという遺伝子のコード化である。

コーディングにおいては、当然のことながら解と成り得る可能性のあるようなものをすべて網羅し、また、定義域から外れたり、評価値が低くその遺伝子が次世代の遺伝子に全く影響を与えないようなものをできるだけ排除したコーディングが望ましい。なぜならば、そうすることによつて多くの場合、より早くより無駄なく収束させることができるからである。しかし、前述のポイント以外にもコーディングにおいて考慮すべき重要な事柄が存在する。

3 本研究でのモデル

3.1 山登り問題

ある閉区間において、ある関数 $f(x)$ が最大となる x を見つける。

この問題を山登り問題と呼ぶことにし、このモデルにおいて実験する。山登り問題を Genetic Algorithm によつて解くには、ある x 座標 ($0 \leq x \leq 255$) を個々の個体として、コーディングして遺伝子とする。遺伝子は 0 または、1 のビット列で、8 ビットの固定長で各個体を表す。また、適合度関数はその x 座標からその位置における高さを返す関数となる。

3.2 Genetic Algorithm の問題点

この山登り問題を図2のような適合度関数において行なう場合はうまくいかない。この関数ではランダムサーチと同程度にしか収束せず、Genetic Algorithm として

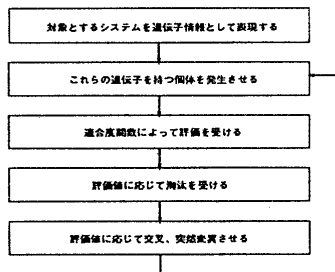


図 1: Genetic Algorithm

2 GA の遺伝子コーディング

GA における遺伝子コーディングとは、GA が扱う進化させるべき色々な個体や、状態をどのようにして遺伝

の効果を得られないのである。

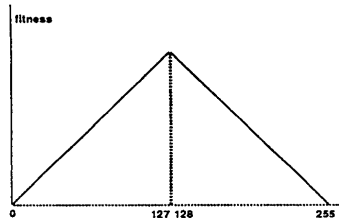


図 2: Genetic Algorithm の不得意な例

その理由は、通常 Genetic Algorithm は x を 2 進コード化し、それを遺伝子として扱っている。図 2 のような適合度関数で GA を実行したとき、なかなか収束せず解らしきものを得るのに多くの世代と時間を必要とする。その理由は図 3 のように、この関数ではすべての第 7 ビットが 0 の遺伝子の評価の和と、すべての第 7 ビットが 1 の遺伝子の評価の和が等しいのである。つまり、第 7 ビットは 0 も 1 も同定度に継承されていき、他方と比べて圧倒的な優勢になるには長期にわたる世代交替が必要である。さらに、第 0 ビットから第 6 ビットでも第 7 ビットと同様に評価の和は等しい。したがって、すべてのビットにおいて優劣が存在せずある特定の解になかなか収束しないのである。図 2 の上では中央に高適応度個体があるように見えるが、それらの遺伝子は似ていないためなかなか収束しないのである。

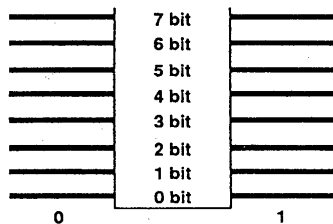


図 3: ビット毎の遺伝子の評価値の和

4 提案

GA を行なうには、より正確な解により高速に収束するような遺伝子コーディングを用いて実行させたい。しかし、コードの善し悪しは GA をおこなうとき簡単に分かるものではない。適合度関数はどのような関数であり、どれほど複雑であるのかは問題より毎回変化し、関数の特徴を完全に把握してから GA を行なうわけではないからである。

そこで、遺伝子コードを動的に変化させることができる機能を持った GA を提案する。これは、何世代か実行した結果から、コーディングがうまくいってないと判断した場合、GA 内部でコーディングを変化させてそれ以後の無駄な世代を作るといった繰り返しを行なわないようにするのである。

5 結論

どんな関数にでも適応し優れた効果を出す万能のコーディング法は存在しないであろう。つまり、コーディングは関数に依存していると考えられるので、それぞれの関数別に特有の最適コード化が存在するのである。したがって、そのコーディングをどのように見つけ、そのコード化の善し悪しをどのように正確に判断するかは人の技術に頼っていた。そこで、最適コード化を自動的に見つけるように Genetic Algorithm を機能拡張したことにより、収束するまでの最長世代数を軽減することができた。

参考文献

- [1] Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning* Addison-Wesley, 1989.
- [2] Holland, J. H. *Adaptation in natural and artificial systems* The University of Michigan Press, 1975.
- [3] Grefenstette, J. J. et al. *Genetic algorithms for the traveling salesman problem* Proc. of an International Conference on Genetic Algorithms and Their Applications, pp.160-168.