

進化的アルゴリズム

八神 翔太, 桑島 奨

Shota YAGAMI, Sho KUWAJIMA

1 はじめに

人が生活していく上で、数学式で答えを出せない問題が数多くある。例えば、速度を上げるために、より空気抵抗の少ない形状を目指した新幹線がある。しかしその形状の作り方は数多くあり、数学式で導き出すには時間がかかりすぎた。そこで、遺伝的アルゴリズムを用いて、その形状を最適化した。その結果、わずか 3 年で N700 系の仕様を決定し、旧型 700 系に比べて空気抵抗が 20 % 削減された。これらの問題には最適な解を導くアルゴリズムは存在するが、最悪の場合、全ての事象を調べることになる。しかし、この場合、最適な解は得られるが時間がかかる。そこで、最適な解をいかに短時間で導き出すために進化的アルゴリズムが提唱された。

2 進化的アルゴリズム (Evolutionary Algorithm : EA)

2.1 概要

進化的アルゴリズムとは、生物の集団遺伝・進化の過程を模倣して最適解を探索するための探索モデルあるいはアルゴリズムの総称である。この手法は、問題に対して変化と選択に基づく世代交代を繰り返すことで解の集団を進化させ最適解を得る。「遺伝子の数字列表現」と「適応度評価関数の設定」が出来れば、幅広い問題に適用出来る。しかし、1 回の探索が短時間で出来る問題でも、全ての解の数が膨大な問題は計算自体に時間が掛りすぎて、有効な解に到達出来ないことも起こり得る。

2.2 EA の分類

遺伝的アルゴリズム, 進化的戦略, 進化的プログラミングは 1990 年まで、独立した研究であった。1990 年代初め、これらが 1 つの技術を表していることが明らかとなり、総称して進化的アルゴリズムと呼ばれるようになった。また、同時期に新たな手法としてとして遺伝的プログラミングが登場した。

- 遺伝的アルゴリズム (Genetic Algorithm : GA)

1975 年に Holland により適応的探索アルゴリズムとして提唱され、EA の中で最も一般的なアルゴリズムである。進化をモデルとした確率的発見の手法の一つで、対象とする問題を環境、対象とする問題の解を個体とする。次に個体ごとに遺伝子という情報を与える。これをコード化とよび、遺伝子がもつ情報を 0, 1 などの bit 表現で表す。全ての個体はコード化された遺伝子情報を持っている。GA では、異なる遺伝子情報を持った個体を集合することで、初

期集団を生成する 環境に対して個体がどれくらい適応しているのかの指標となる適応度が与えられ、選択/淘汰によってより高い適応度を持つ個体を多く残す。選択とは、環境において適応度の高いものの割合を増やすことである。淘汰とは、環境において適応度の低いものの割合を減らすことである。次に交叉や突然変異により多様な個体を生み出す。交叉とは、遺伝子の一部を、他の個体の遺伝子と入れ換える操作である。突然変異とは、個体の遺伝子の一部を変化させる操作である。突然変異は、個体群の多様性を維持するために行うので、設計変数間に相関がある場合、良好な個体を生成することができない。この過程を繰り返し、解を進化させることで最適解を導くことができる。この繰り返しは個体群の中の最大適応度が、設定された閾値を超える、あるいは繰り返しの回数が設定した回数を超えるなどの条件によって終了する。遺伝的アルゴリズムの過程を Fig. 1 に示す

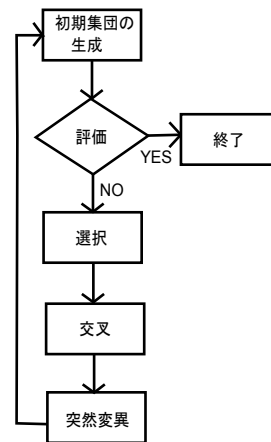


Fig.1 遺伝的アルゴリズムの過程

GA はほかの 3 つの手法に比べて、自由度が高く、設計変数の多い大域的な最適化に有効である。10⁴ (10~20) 程度の解を持つ大きさの問題に有効である。広範囲における探索、最適化問題、機械の学習問題など多岐にわたって応用可能である。一方、繰り返し計算による計算負荷の大きさ・早熟収束により局所解へ収束する可能性の 2 つの課題点がある。早熟収束は、最初の方の世代で他の個体より適応度が圧倒的に高い個体が生まれた場合に起こる。その個体の遺伝子が集団中に爆発的に増えて、解の探索が早い段階で収束してしまう現象である。

- 遺伝的プログラミング (Genetic Programming : GP)

GP は 1987 年に Koza により提案されたものである。GA の応用である。GA との違いは遺伝子の表記のみ異なり、アルゴリズムの過程は同一である。遺伝子を木構造にすることで式やプログラムなどを扱えるようにした。これより遺伝子の構造が 1 次配列状になり、どの部分に対して遺伝子操作が行われたかが視覚化される。安定なアルゴリズムを実現するには、環境の変化により影響されることを阻止しなければならない。しかし、その影響を正確に予想することは容易でない。特に数式やプログラムのコードの表現は重要な要因を把握し、扱うのが容易ではない。視覚化されることで数列表現に比べて、設計がイメージしやすくなる。結果、より環境の変化による影響に強い設計を可能にする。適応度関数は計算能力で評価される。一方で、最適解を見つけない場合、木構造が深く、そして複雑になるため 1 つの遺伝子にかかる計算が大きくなる。

- 進化的戦略 (Evolution Strategy: ES) ES は非線形最適化問題を解くことを目的に、1965 年、Rechenberg と Schwefel により開発されたアルゴリズムである。GA は交叉手法を用いて遺伝子が持つ情報を交換し、解を繰り返し混同させるのに対して、ES は突然変異と選択手続きだけを用いる。交叉を用いないため、交叉の設計により局所解に陥ることが少ない。最適化する解が多数の局所最適解をもつ場合でも適応可能であるという適応範囲の広さにある。進化的戦略の過程を Fig. 3 に示す。

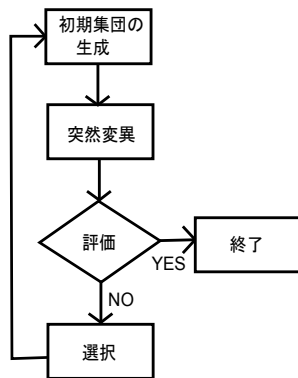


Fig.2 進化的戦略の過程

- 進化的プログラミング (Evolutionary Programming: EP) EP は予想問題を解くための有限状態機械を進化させる枠組みとして、1965 年、Fogel により開発されたアルゴリズムである。EP は他の 3 つの方法論と比べると、特に決まった構造が無い。個体がお互いに、次の世代に選択される過程を直接計算する。そのため、低い適用度を持つ個体は、もしあいての個体がさらに低い適用度を持っていれば選択される可能性が高まる。

3 実用例

3.1 日除け

京都大学において遺伝的アルゴリズムを用いた「日除け」の開発が進められている。都会は郊外に比べて暑いと言われているが気温には違いがなく、地表面の温度が高いため都市部の方が暑く感じるというデータがある。そこで、地表面の温度を下げるために自然界にある木陰を模倣した日除けの開発が勧められた。屋根のように日光を完全に遮断するのではなく、隙間の多い構造から適度に日光が漏れるようになっている。これにより日除け自体の温度が上がらず、日除け自体が熱を放出することがない。実際に普通の屋根と比較するとその温度は 20℃ 以上低くなっている。影を作ることで地面の温度が上がりにくくなり、日除けの風通しを良くすることで空気が籠りにくくなっている。結果、熱くなりにくい。シミュレーションで気候条件や日射量など、条件を変えつつ、日除けの形を遺伝的アルゴリズムで変化させれば、より良い日除けを作ることができる。

3.2 太陽光発電装置の設置

太陽光発電装置の設置における問題点で大きな面積の土地が必要であることが挙げられる。事実、家庭で設置するのにも屋根以外の設置場所は少ない。そこで葉を光合成する樹木のようにできるだけ太陽光パネルに日が当たるように設置することで、同じ面積の土地で多くの発電をすることを目指している。遺伝的アルゴリズムを用いることによって最適な設置を導き出すことが可能になる。

4 まとめ

すべての問題に万能なアルゴリズムは存在しないという事実がある。各々のアルゴリズムにはそれぞれ長所短所があり、問題によって、アルゴリズムの効果が異なる。EA は分類される 4 つの手法の多様性を維持しつつ相互に影響を及ぼしながら発展してきた。今後もこのように発展していくであろうといえる。

5

参考文献

- 1) 松本啓之亮, 黄瀬浩一, 森直樹. 知的システム工学入門. 2002.
- 2) 電気学会 GA ニューロを用いた学習とその応用調査専門委員会. 学習とそのアルゴリズム: ニューラルネットワーク・遺伝アルゴリズム・強化学習. 2002
- 3) ロルフファイファー, クリスチャンシャイアー, 石黒章夫, 細田耕, 小林宏. 知の創成—身体性認知科学への招待. 2001
- 4) フラクタル日除け— 京都大学.
<http://www.gaia.h.kyoto-u.ac.jp/~fractal/index.html>.
- 5) 進化的計算.
<http://www.eva.ie.u-ryukyu.ac.jp/~endo/classes/Gaintro.pdf>.