生物学特論A
(分類系統学II)
第12回

遺伝的アルゴリズム

遺伝的アルゴリズム(Genetic Algorithm; GA)とは，生殖，突然変異，遺伝子組み換え，自然淘汰，適者生存，などという仕組みが用いた，最適化計算の一種である。生物進化に着想を得た操作であることから進化論的計算と呼ばれる。

最適化計算手法として

最適化問題の解の候補群が生物の個体群の役割を果たし，コスト関数によってどの解が生き残るかを決定する。生き残った個体で遺伝子組み換え（交配），突然変異が起こり，個体群の進化が行われる。この操作を繰り返すことで最適化問題に対する解を得ようとする計算論的手法である。

GA の適用範囲は広く，工学，芸術，生物学，経済学，遺伝学，オペレーションズリサーチ，ロボット工学，社会科学，物理学，化学などの分野で応用されている。

遺伝的アルゴリズム（GA）は，1975年にミシガン大学の John Henry Holland によって提案された近似解を探索するメタヒューリスティックアルゴリズムである。人工生命同様，偶然の要素でコンピューターの制御を左右する。GA はデータ（解の候補）を遺伝子で表現した「個体」を複数用意し，適応度の高い個体を優先的に選択して交叉（組み換え）し，突然変異と交配の操作を繰り返しながら解を探索する。適応度は適応度関数によって与えられる。この手法の利点は，評価関数の可微分性や単峰性などの知識がない場合であっても適用可能なことである。 必要とされる条件は評価関数の全順序性と，探索空間が位相（トポロジー）を持っていることである。また，遺伝子の表現の仕方によっては組合せ最適化問題やNP困難な問題などのさまざまな問題に適用可能である。

一般的な計算の流れ

GA は進化と自然淘汰の仮説の正当性を実験検証するのにも使われてきた。GA を用いることで，おどろくほど短時間の繰り返しで解が求まるからである。ただし，GAは一般に小進化に限定される（大進化のシミュレートする試みもなされている）。

GA への批判としては，遺伝子型と表現型の区別が不明確である点が挙げられる。実際，受精した卵細胞は胚発生という複雑なプロセスを経て円熟した表現型になる。この点を実現したアルゴリズムは少ない。これが実現されれば生物の進化可能性も改善されると考えられる。人工胚発生や人工発生システムの研究では，これらの懸念への対処が行なわれている。

一般的な計算の流れ

1. 解の候補を染色体（遺伝子）として表現する
2. どの染色体（遺伝子）を選ぶかを決める
3. 交叉（交配）相手を選ぶ
4. 突然変異を起こす確率を決め適用する
5. 次世代の個体数を決める
6. 上記の計算（世代回数）を繰り返す

もう少し具体的な計算の流れ

1. 個体数 N 個が入る配列を二つ用意する。この二つの集合を「現世代」，「次世代」と呼ぶ。
2. 現世代に N 個の個体をランダムに生成する。
3. 評価関数により，現世代の各個体の適応度を計算する。
4. ある確率で次の3つの動作のどれかを行い，その結果を次世代に保存する。
   1. 個体を二つ選択して交叉を行う。
   2. 個体を一つ選択して突然変異を行う。
   3. 個体を一つ選択してそのままコピーする。
5. 次世代の個体数が N 個になるまで上記の動作を繰り返す。
6. 次世代の個体数が N 個になったら次世代の内容を全て現世代に移す。
7. 3. 以降の動作を最大世代数 G 回まで繰り返す。求める解は「現世代」の中で最も適応度の高い個体である

個体の表現

実際の生物では，染色体は，デオキシリボ核酸 DNA によって構成されており，アデニン，シトシン，グアニン，チミンと呼ばれる塩基である。すなわち4通りの可能性があって，遺伝子の長さが m であれば 4^m とおりの表現が可能である。GA では，最適化すべき対象をベクトル表現して

$$\mathbf{x} = \{x_1, x_2, \cdots, x_m\} \qquad (4)$$

と表現する。生物学からのアナロジーにより，x_i が取りうる値を対立遺伝子，各遺伝子が入る染色体の場所を遺伝視座と呼ぶ。

個体の適応度

進化論的考え方に従えば，各個体は環境への適応度がそれぞれ異なり，適応度に応じて自然選択される。適応度の高い個体は次の世代へ子孫を残しやすく，反対に適応度の低い個体は子孫を残すことが困難になる。GA ではこの考え方を取り入れて，各個体に適応度 f を付加する。最適化すべき目的関数 g(x) が最大となるような適応度を持つ個体が選ばれる。

遺伝的操作

遺伝的アルゴリズムでは一般的に次の遺伝的操作が用いられる。

1.選択（淘汰，再生）
2.交叉（組み換え）
3.突然変異

交叉する確率を交叉率，突然変異する確率を突然変異率という。一般には（交叉率）>>（突然変異率）とすることが望ましいとされる。また上記のアルゴリズムの流れからわかるとおり

交叉率 + 突然変異率 < 1

である必要がある。

選択

選択は生物の自然淘汰をモデル化したもので，適応度にもとづいて個体を増やしたり削除したりする操作である。 選択のアルゴリズムには次のようなものがある。

1. ルーレット選択
2. ランキング選択
3. トーナメント選択
4. その他

ルーレット選択

ルーレット選択とは，個体 i の適応度をf_i とし，個体 i の選択確率を p_i としたとき，次式，

$$p_i = \frac{f_i}{\sum_{k=1}^{N} f_k} \qquad (5)$$

で選択する方式である。例えば，以下のような表に従って選択する

ルーレット選択(2)

ルーレット選択の例

個体	適応度	選択確率
1	10	0.1
2	60	0.6
3	30	0.3

この方式はホランドが最初に提案したときに使われた選択方式であり，最も有名な選択方式である。ただし，適応度に負の数があるときには使えないので，全適応度正の値に変換するなどの工夫が必要。さらに，各個体の適応度の差が著しい場合，適応度の高い個体のみが選択される確率が高くなり，局所最大に収束してしまい，最適解を求められない場合がある。

ランキング選択

ランキング選択は各個体を適応度によってランク付けして，「1位なら確率 p_1, 2位なら確率 p_2, 3位なら...というように，ランクごとにあらかじめ確率を決めておく選択方式である。この方法は，ルーレット選択と違い選択確率が適応度の格差に影響されない。しかし，これは逆に適応度にあまり差がない個体間でも選択確率に大きな差が生じる可能性がある。

トーナメント選択

トーナメント選択はあらかじめ決めた数（トーナメントサイズという）だけ集団の中からランダムに個体を取り出し，その中で最も適応度の高い個体を選択する方式。トーナメントサイズを変更する事で選択圧をコントロールできる特徴がある。すなわち，トーナメントサイズを大きくする事で選択圧を高める事ができる。トーナメントサイズを大きくすると，そのなかで最大の適応度を持つ個体だけが選ばれることになり，適応度の低い個体が生き残ることが難しくなる。

その他

上記の選択とは別に適応度が高い個体（エリート）を一定個数，次世代に残すことがある（エリート選択）。これを利用することで，選択によって解が悪い方向に向かわない（適応度の最大値が下がらない）ことを保証できる。しかし，エリートの遺伝子が集団の中に広まりすぎて解の多様性が失われるという恐れもある。

交叉（組み換え）

交叉（組み換え）は生物が交配によって子孫を残すことをモデル化したもので，個体の遺伝子の一部を入れ換える操作である。交叉はその性質上，最も重要な遺伝的操作と言うことができる。交叉のアルゴリズムには次のようなものがある。

例として次の二つの個体を交叉する。

個体A: 0100111010
個体B: 1010101011

一点交叉

遺伝子が交叉する場所（交叉点）をランダムで一つ選び，その場所より後ろを入れ換える方式である。ホランドが最初に提案したときの交叉方法であるが，効率は低く現在ではあまり使われていない。

個体A: 01001｜11010 ⇒ 01001 01011
個体B: 10101｜01011 ⇒ 10101 11010

二点交叉

交叉点をランダムで 2 つ選び，2 つの交叉点に挟まれている部分を入れ換える方式である。

個体A: 010 | 01110 | 10 ⇒ 010 01010 10
個体B: 101 | 01010 | 11 ⇒ 101 01110 11

多点交叉

一般に，3点以上の交叉点をもつ方法は多点交叉あるいは n 点交叉と呼ばれる。しかし，多点交叉は二点交叉と下記の一様交叉のどちらかよりも良い値が出ることはなく，あまり使われない。

一様交叉

各要素ごと独立に 1/2 の確率で入れ換える交叉である。後述するヒッチハイキングの問題をおさえることが可能である。一般に二点交叉が得意とする問題を苦手とし，二点交叉と逆の性質を示すことが知られている。

個体A: ０１００１１１０１０⇒００１０１１１０１１
個体B: １０１０１０１０１１⇒１１００１０１０１０

突然変異

突然変異は生物に見られる遺伝子の突然変異をモデル化したもので，個体の遺伝子の一部を変化させる操作である。局所（的）最適解に陥ることを防ぐ効果がある。ランダムに選んだ任意の遺伝視座において，遺伝子の値を対立遺伝子に置き換えることをさす。たとえば対立遺伝子が {0,1} であるとき

突然変異

１ ０ １０

の2 番目の遺伝視座に変異が起きたとすれば

１ １ １０

などとなる。対立遺伝子がn 個の整数で表現されていれば，0 からn までの乱数を使って置き換えるなどの操作をする。
突然変異の確率は0.1%〜1% 程度であり，高くても数%である。確率が低すぎると局所最大値に陥って抜け出せなくなり，高すぎるとランダム探索になり解の探索効率が落ちてしまう。

ヒッチハイキング

例えば最適解が

- １ ０ １ -

と言う問題があるとする。このとき

- １ １ １ -
- ０ ０ ０ -

という二つの個体が交叉して最適解を得る確率を求める。 交叉の方式が二点交叉の場合は交差点が

-1|1|1- ⇒ -101-
-0|0|0- ⇒ -010-

ヒッチハイキング(2)

で最適解が得られる。このとき遺伝子型の長さを l とおくと，最適解が得られる確率 p は，

$$p = \frac{2}{l(l-1)} \qquad (6)$$

と求められる。これは l が長くなるにつれ加速度的に確率が低くなる。つまり l が長いとほとんどの確率で上記の 2 つの個体は最適解と一致しないビットを新しく生成した個体に受け継がせてしまうことになる。このように最適解と一致するビットの近くにいて最適解の生成を妨げる現象をヒッチハイキングといい，そのビットをヒッチハイカーという。

ヒッチハイキング(3)

このヒッチハイキングは一様交叉によって防ぐことができる。一様交叉は各要素が独立で交叉するので，上記の場合は

-111-⇒ -101-
-000-⇒ -010-

か

-111-⇒ -010-
-000-⇒ -101-

で最適解を得る。このとき，最適解を生成する確率は

ヒッチハイキング(4)

$$p = \frac{2}{2^3} = \frac{1}{4}$$

であり，この確率は l の長さが長くなっても変化しない。

SGA

SGA とは Simple Genetic Algorithm（単純 GA）の略。GAを通常のまま解析するとあまりにも複雑なので，処理を単純にした GA を用いて解析を進めるのが一般的になっている。SGA は具体的には

1.遺伝子表現は 1 と 0 のみ

2.ルーレット選択

3.一点交叉

4.突然変異は1箇所の遺伝子座の値を反転させる

という実装のアルゴリズム

実習

今回の実習プログラムは，Perl と呼ばれるスクリプト言語で書かれている。

./SGA.pl

とタイプすれば，シミュレーションが始まる。

実習

SGA.pl には，オプションを6つ指定することができる。それぞれ，

-l : 遺伝子長（デフォルトでは20）

-p : 個体数（デフォルトでは10）

-m : 突然変異率(デフォルトでは 0.01)

-c : 交叉率(デフォルトでは 0.01)

-g : 世代数(デフォルトでは 100)

-s : 乱数の種

いろいろとオプションで指定できる数値をいじって遊んでみよう。

文献

- 伊庭斉志. (1994). 遺伝的アルゴリズムの基礎. 東京: オーム社.
- 伊庭斉志. (2002). 遺伝的アルゴリズムと進化のメカニズム. 東京: 岩波書店.
- 梅原嘉介, & 小川敬治. (2007). 進化ゲームと遺伝的アルゴリズム. 東京: 工学社.