

河村 優†

小林 哲二‡

松田 郁夫§

日本工業大学

1 はじめに

本研究は、遺伝的アルゴリズム^{III} (Genetic Algorithms, 以下 GA) の手法を取り入れた、プログラムの自動生成に関するものである。プログラムの基本要素の関数を遺伝子と考え、その遺伝子列の順番が一つの機能を表すと考え、GA オペレータを用いて可変長の遺伝子列の順番や組み合わせを変異させ、簡単な局所的規則により、目的とするプログラムの自動生成を行うシステムの構築を目指している。

2. 基本的な考え方

プログラムは有限要素の機能の組み合わせにより構成されている問題解決のための手順と考えることができる。

そこで、局所的規則を用いて除外できる遺伝子列を除去するのに、総当たりの解決法ではなく、ヒューリスティックな解決法として GA を用いる。

ここでは、簡単な例として、水差し問題を取りあげ、問題解決のための個々の基本動作を複数個のルールとして表現し、ルールの列を遺伝子列として考えることにした。

3. 水差し問題

例題とする水差し問題の概要を述べる。与えられた目盛りの振っていない二つの容器 A, B に入っている水の量をそれぞれ a , b リットルとする。 a , b それぞれの最大値をそれぞれ a' , b' ($a' > b'$)

リットルとする。これらを使って x ($x < b'$) リットルの水を計量する。ただし a , b , x は整数とする。

初期状態 ($a = 0$, $b = 0$)

最終状態 ($a = *$, $b = x$)

ただし、 $*$ は $0 \leq * \leq a'$ なる整数とする。

水差し問題を解くための基本動作をルールとして記述したものを表 1 に示す。このルールは、現状態に依存せずに使用できるような記述とした。そのためルールの列は意味の無い無駄な動作を行う可能性もある。またその際に考えられるルールの適用可能数は、重複をゆるした順列となるので、その総数 m はルール列の長さを n とすると (1) 式で与えられる。

$$m = 6^n \cdots \cdots (1)$$

表 1. 水差し問題のルール

ルール番号	作業内容
1	A に水を満たす
2	B に水を満たす
3	A の水を捨てる
4	B の水を捨てる
5	A から B にできる限り水を移動
6	B から A にできる限り水を移動

初期状態より到達可能な状態に到達するルール列の長さおよびルールの適用順番について GA を用いて発生させる。最終状態に到るルール列の発生を目的とする。

ルール 6 を例として、LISP を用いて関数化したものを以下に示す。

```
(defun Rule6 (NowA NowB MaxA MaxB)
  (let ((Move (- MaxA NowA)))
    (if (<= NowB Move)
      (list (+ NowA NowB) 0)
      (list (+ NowA Move) (- NowB Move)))))
```

* Program auto-generation by using GA.

† Masaru KAWAMURA

‡ Tetsuji KOBAYASHI

§ Ikuo MATSUDA

Nippon Institute of Technology 4-1 Gakuendai,
Miyashiro-Machi Minimisaitama-gun, Saitama-ken
345-8501 JAPAN

3-1 水差し問題の解の存在

水差し問題の解は a' , b' の最大公約数が求める x の約数でなければ存在しないことが知られている。よって解が存在する為には, (2) 式が成立しなければならない。

$$x \bmod \gcd(a', b') = 0 \quad \cdots \cdots (2)$$

これは, 大域的な制約なので, 問題解決には, 直接用いることは, できないが, 問題の前提条件と考える。

4. 遺伝的アルゴリズムの適用

4-1 遺伝子構造

表 1 に示した水差し問題の各ルールを 1 個の遺伝子とし, 複数のルールで構成されるルール列が 1 個の遺伝子構造を形成する。その長さ α は可変長とする。図 1 に遺伝子構造を示す。図中にある数字は対応するルールの番号を表している。

ある長さの遺伝子構造からスタートして, 目的となる解が得られるまでその長さを順次長くしていく。また遺伝子の最大長 α' は, a' , b' の最大公約数を求める手順数 β , 求める x を基準に算出することは可能であるが, このような大域的値は用いずに適当な長さのものをを用いる。

1	5	4	2	1
---	---	---	---	---

図 1. 遺伝子構造

4-2 使用する GA オペレータ

本研究で用いる GA オペレータは, 標準的に用いられる突然変異, 逆位などでその適用例は, 図 2, 3 に示す。

1	5	4	1	5
↓				
1	5	4	3	5

図 2. 突然変異の適用例

1	5	4	1	5
↙ ↘				
1	5	1	5	4

図 3. 逆位の適用例

4-3 適合度について

GA の適合度の計算は, この後で述べる局所的規則と遺伝子長から決定する。遺伝子長が短いほど適合度を高く, 長いほど低くなるように設定する。

5. 局所的規則

無作為に変異を繰り返したのでは, 目的とするプログラムの生成までに多くの時間がかかってしまう。そこで, 簡単に, 局所的な規則を導入することにより, 試行回数の増大を防ぐ。

局所的規則の例

- ・ 以前の状態に戻る
- ・ 意味の無い動作を繰り返す

このような局所的規則に当てはまった遺伝子の組み合わせは, 死滅させる。

6. システムの流れ

はじめに, 初期パラメータである a' , b' , x を入力する。遺伝子列の生成と適合度の計算を繰り返すことにより, 目的となる遺伝子列を求める。

7. おわりに

本研究では, 求める解が一定の操作の組み合わせで, 求められるものについてのみ考えている。今後の課題としては以下のようなものが考えられる。

- ・ 遺伝子列を, 途中で切り離したり, 結合する動作を行うオペレータの導入。

参考文献

- [1] 伊庭斉志: “遺伝的アルゴリズムの基礎 —GA の謎を解く—”, オーム社, 1995