

アルゴリズム論(第12回)

2004.1.19

櫻井 彰人

algorithm@soft.ae.keio.ac.jp

http://www.sakurai.comp.ae.keio.ac.jp/

きょうの講義概要

■ アルゴリズムについての話題の補足

- 遺伝的アルゴリズムなど

» 「アルゴリズム」らしくないアルゴリズム

7らしくないアルゴリズム

◆ 確率的

- 確率的アルゴリズム

- 複雑系

- 遺伝的アルゴリズム

◆ 新しいコンピュータ

- 並列処理や分散処理

- 量子コンピュータ

遺伝的アルゴリズム 概要

◆ 開発: 米国 1970年代

◆ 開拓者: J. Holland, K. DeJong, D. Goldberg

◆ 主たる応用:

- 離散最適化

◆ 性格:

- 余り速くない

- 組合せ最適化に役立つヒューリスティクス

◆ 特徴:

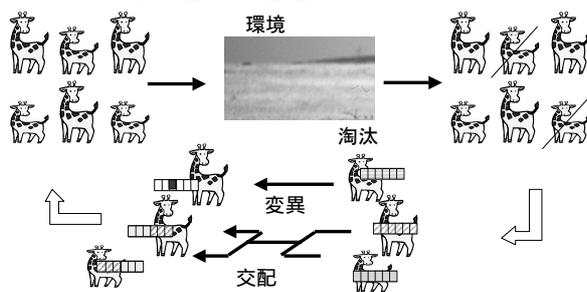
- 伝統的に言われるのは: 良い準解を組み合わせでよりよい解を作る (染色体交叉)

- 変種多し, e.g., 新たな解候補を作る演算、個体の管理

遺伝的アルゴリズム

生物の進化の過程を模倣

◆ 生物の進化の過程



遺伝子と進化

◆ ゲノムと遺伝子

- ゲノムとは、ある生物がもつ遺伝情報の全体

- ゲノムのなかで生命活動を維持するための機能的な部品を規定しているところが遺伝子

- 染色体上のDNAが遺伝情報をもっている。従ってDNAの一部、塩基対の列が遺伝子

» DNAは約30億個の塩基対

» <http://www.genome.ad.jp/Japanese/tutorial/01-01.html>

» http://www.aist.go.jp/aist_j/dream_lab/mame/01.html

- 進化

» 細胞内の遺伝子の組換え

» 細胞内の遺伝子の突然変異

» 細胞間の遺伝子の入れ替え

遺伝子とは生物の体の形や働き(形質)の特徴を規定するものです。
 人間は3~4万の遺伝子を持っていますが、そのほとんどの遺伝子は全ての人に共通です。ごく少数の遺伝子の違いが、顔かたちや体質の違いとなって現れます。

遺伝子の本態はDNAという物質です。このDNAの鎖が大量に集まってできる染色体が全ての細胞内にあります。

細胞
染色体
DNA鎖

<http://www.kuhp.kyoto-u.ac.jp/idennet/idensoudan/illust/apb-ic02.ppt>

遺伝情報の伝達と変化

◆ 染色体の交叉

個体1の染色体
0 1 0 0 1 1 | 0 1 1

個体2の染色体
1 1 1 0 1 1 | 1 0 1

0 1 0 0 1 1 1 0 1

遺伝情報の伝達と変化(2)

◆ 突然変異
- 遺伝子の一部が変化

0 1 0 0 1 1 0 1 1

↓

0 1 1 0 1 1 1 1 1

遺伝的アルゴリズム

生物の進化の過程を模倣

◆ 模倣

準解1 準解2 → 良いところを併せる → 準解3

良いもの → 組み合わせる → 進化

表現と演算

表現型 phenotype 空間

遺伝子型 genotype 空間 = {0,1}ⁿ

符号化

復号化

淘汰・適者生存

遺伝子の変化(変異と交叉)

遺伝的アルゴリズム

- ◆ Genetic algorithm, 略称 GA
- ◆ Holland のオリジナル版は現在では simple GA (SGA) と呼ばれる

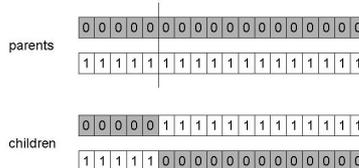
表現	ビット列
組み換え	N-点 or 一様
変異	ビット反転 固定確率
親の選択	適合率比例
生存者の選択	親は子供で入替え
強調	交差

SGA 増殖サイクル

1. 個体を集めて集団 (population) とする
2. 2個ずつ対をつくる(ランダムに)。
3. 交叉を用いて子供を作る(交叉なしもあり)
4. 子供遺伝子に、突然変異を起こす
5. 集団全体を子供で置き換える

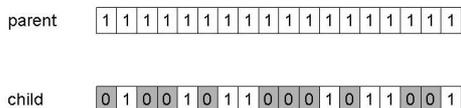
SGA 操作: 1点交叉

- ◆ 親遺伝子の一点をランダムに選択
- ◆ この点(交叉点)で親遺伝子列を二つに分割
- ◆ 尻尾部分を入替えて子供を作る



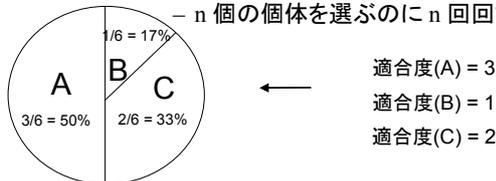
SGA 操作: 突然変異

- ◆ 遺伝子それぞれを一定確率 p_m で変化させる
- ◆ p_m は突然変異率

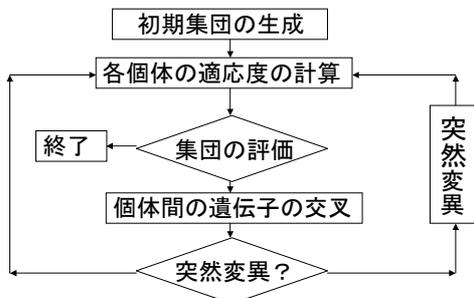


SGA 操作: 選択

- ◆ 主アイデア: 適した個体は高い生存確率
 - 適合度 (fitness) に比例した確率
 - 手法: ルーレット板
 - ルーレット板の部分を各個体に割当
 - n 個の個体を選ぶのに n 回回す



遺伝的アルゴリズム



例: Goldberg '89 より

- ◆ 単純な例題: x^2 の $\{0,1,\dots,31\}$ における最大値
- ◆ GA を適用するには:
 - 表現: 2値表現, e.g. 01101 \leftrightarrow 13
 - 集団の大きさ: 4
 - 1点交叉, ビットごとの突然変異
 - ルーレット選択
 - 初期値はランダム
- ◆ 1世代分を例示する

x^2 例: 選択

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

x^2 例: 交叉

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 1	4	0 1 1 0 0	12	144
2	1 1 0 0 0	4	1 1 0 0 1	25	625
2	1 1 0 0 0	2	1 1 0 1 1	27	729
4	1 0 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

x^2 例: 突然変異

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

単純 GA

- ◆ (初期の) 多くの研究の対象
 - 今でも新 GA のベンチマークに使用
- ◆ 短所もたくさん e.g.
 - 表現への制限がきつすぎる
 - 突然変異 & 交叉はビット列や整数表現にしか適用できない
 - 適合度が近い集団に対しては、選択が敏感に働きすぎる
 - 集団の生成方法は、適合度の高いものを優先するようにすれば、改善される

1点交叉の問題点

- ◆ 1点交叉の性能は、表現ないの遺伝子の並ぶ順序に強く依存している
 - 配置位置が近い遺伝子ほど一緒にいる可能性が高い
 - 染色体の両端にある遺伝子は、一緒にいることはない
 - これは位置のバイアスと呼ばれている
 - 問題の構造が分かっているときは、この性質を活用することができる。しかし、一般には、そのようなことはない

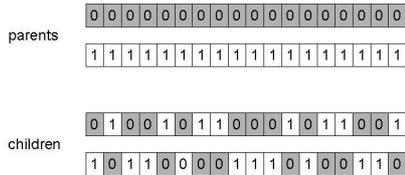
n 点交叉

- ◆ n 点ランダムに交叉点を選ぶ
- ◆ これらの点で分割
- ◆ 各部分を、交互に、糊付けする
- ◆ 1点交叉の一般化(まだ位置のバイアスは残っている)

parents	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
children	0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1
	1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0

一様交叉

- ◆ coin-tossing をする。
- ◆ 表がでたら一方の親から、裏が出たら他方の親から遺伝子をもってきて、子供を作る
- ◆ もう一人の子供は、その裏返し
- ◆ 実は、交叉のよい性質を失っている可能性あり



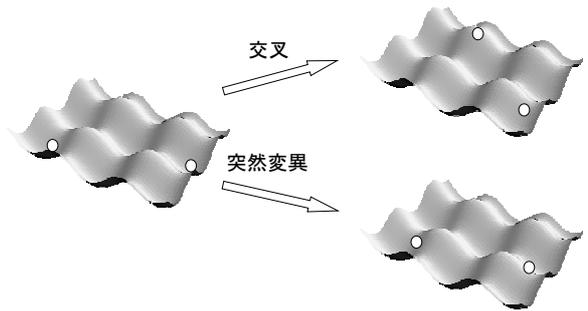
交叉がよいか突然変異か?

- ◆ 長〜い議論: どちらが よいか / 必要か / 主な機能か
- ◆ 応え (とはいえなくも, 同意が得られている):
 - 問題依存, しかし
 - 一般には, 両方あるのがよい
 - それぞれ別の役割
 - 突然変異だけのGAは可能, しかし交叉だけのGAはまずうまくいかない

交叉か突然変異か? (続)

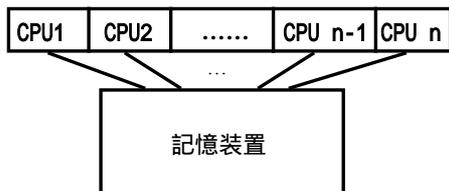
- 探索 exploration: 探索空間内で、うまくいきそうな場所を探す i.e. 問題に関する情報を収集する
- 活用 exploitation: うまくいきそうな場所で最適な場所を見つける i.e. 情報の活用
- 両者間にはトレードオフがある
- ◆ 交叉は探索的, 両親の間近辺の領域に大きくジャンプ
 - ◆ 突然変異は活用的, 親の近くでランダムな小さいステップでの移動を繰り返す

交叉か突然変異か



7.2 並列処理と分散処理

- 並列処理 (Parallel Processing)
 - 複数のCPUに一つのメモリ
 - ベクトルプロセッサが代表的



分散処理

- 分散処理 (Distributed Processing)
 - CPUとメモリの組が複数個
 - 各組どうしは疎な結合

