遺伝的アルゴリズム（Genetic Algorithm ）

生物の進化の過程を模擬して新しい情報処理手法を築こうとするのが遺伝的アルゴリズムです。
ニューロコンピュータが個体としての生物を対象としているのに対して、遺伝的アルゴリズムは何世代にもわたる複数の生物を対象としている点が大きく異なる点。

遺伝的アルゴリズムでは生物進化の特徴である以下の原理を用いる。

淘汰：環境への適応度が低い者はしだいに滅びていく。
増殖：環境への適応度が高い者は栄える。
交叉：性質の混合
突然変異：強制的にわずかに変化した個体を生成する。

具体的には、

１）対象とするシステムの様々なパラメータの集まりを一つの遺伝子とみなして表現する。
２）多くの異なった遺伝子を作って、上の原理を適用していく。

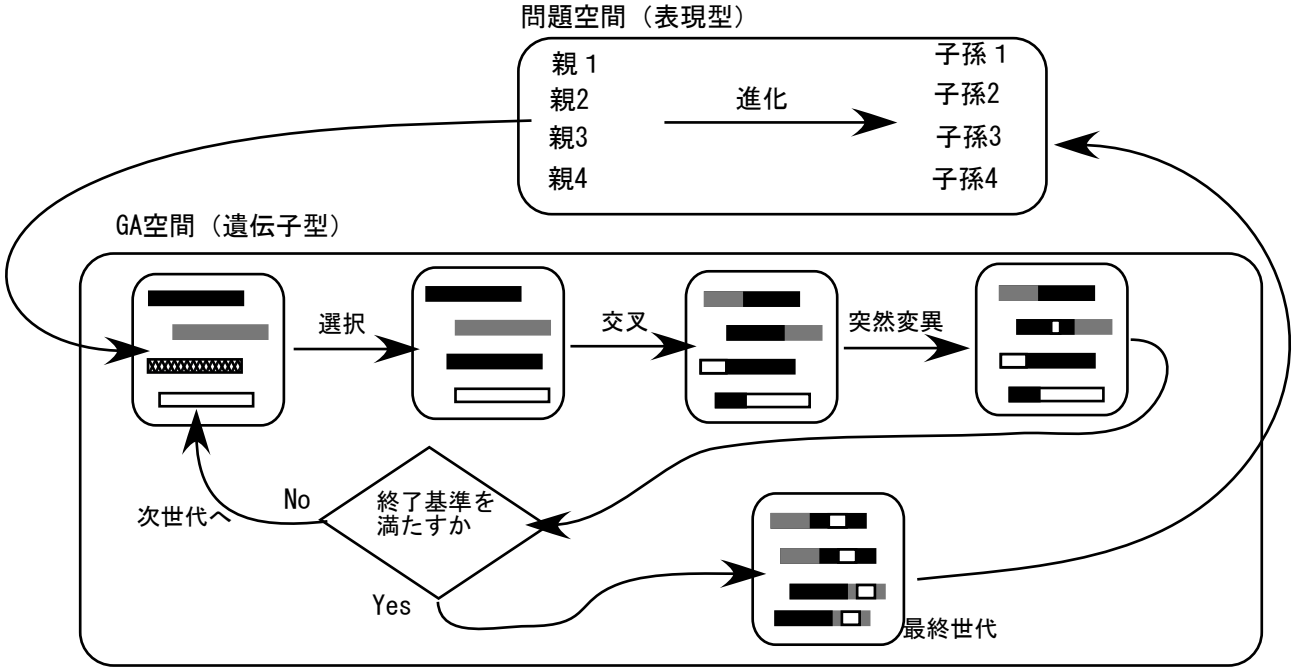
実際の遺伝子

生物は細胞で構成され、その細胞には核があり、その核の中に染色体がある。染色体は主にDNAによって構成されている。このDNAは、４種類の塩基（A:アデニン、G：グアニン、C：シトシン、T：チミン）を主な構成要素としている。DNAのもつ情報はこれらの４種類の塩基の並び方によって規定される。DNAは２重らせん構造をなしていて、これが複雑にたたまれて染色体を構成している。
遺伝子とは、遺伝情報を担うDNAをいう。特定の遺伝子は染色体上の特定の位置（遺伝子座）に存在する。一つの遺伝子座における遺伝子の組み合わせを遺伝子型という。

遺伝的アルゴリズムにおける遺伝子

遺伝的アルゴリズムでは、染色体を１次元に簡単化します。つまり記号や数値を一列に並べて、それを染色体として扱う。また、各位置を遺伝子座として扱う。

遺伝的アルゴリズムの概念図



遺伝的アルゴリズムの一般的な手順

手順１（初期化）

対象とする問題を遺伝子の形にコーディングする。
異なる染色体をもつ個体をN個生成して、初期世代とする。

手順２（選択）

各個体の適合度を計算し、適合度に基づく一定の規則で個体の選択を行う。
一般的に、適合度の高い個体は増殖し、低い個体は淘汰される。

手順３（交叉）

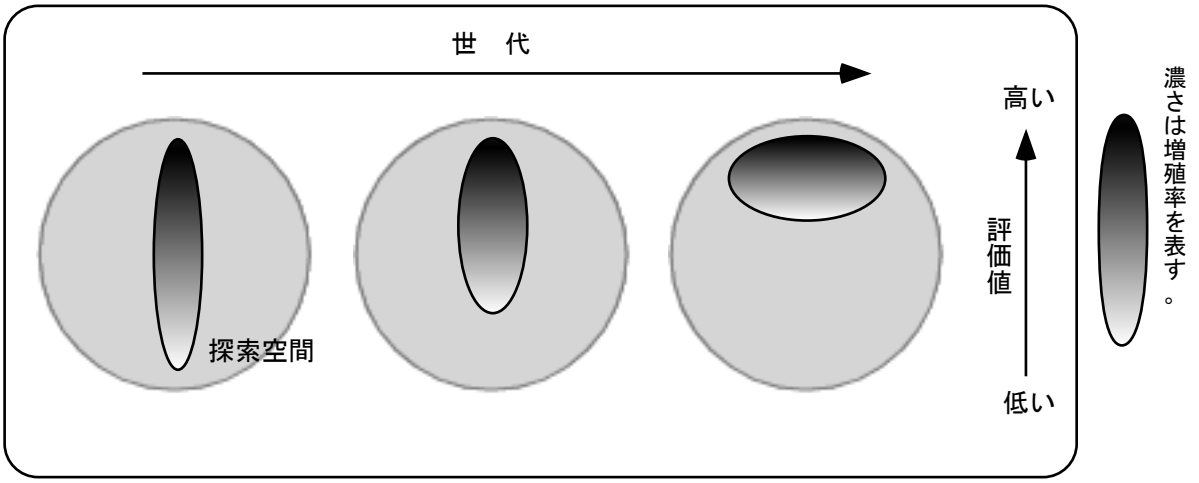
設定された交叉確率や交叉の方法により交叉を行い、新しい個体を生成する。

手順４（突然変異）

設定された突然変異確率や突然変異の方法によって突然変異を行い、新しい個体を生成する。

手順５（終了判定）

終了条件を満たせば、その時に得られている最良の個体を問題の準最適解とする。
そうでなければ手順２に戻る。



図：適合度による進化過程の模式図

遺伝的アルゴリズムの特徴（従来の探索法と比較して）

（１）パラメータをコーディングしたものを直接利用する。
（２）一点探索ではなく、多点探索である。
（３）サンプリングによる探索で、ブラインドサーチである。
（４）決定論的規則ではなく、確率的オペレータを用いる探索である。

遺伝的アルゴリズムの簡単な例題

条件：染色体の長さ→10ビット
適応度→前半5ビットの０の和＋後半5ビットの1の和
選択法→エリート保存選択
交叉法→一点交叉

⇒　最適解→0000011111

第１世代

順位	染色体	適合度
1	10101 11011	6
2	10010 01101	6
3	01011 10110	5
4	00100 01000	5
5	11100 10001	4
6	~~00110 00100~~	~~4~~

淘汰

平均適応度　5.0

増殖

染色体	適合度	順位
10101 11011	6	1
10101 11011	6	2
10010 01101	6	3
01011 10110	5	4
00100 01000	5	5
11100 10001	4	6

平均適応度　5.3

並び替え

順位	染色体	適合度
1	10101 11**011**	6
6	11100 10**001**	4
2	10101 11**011**	6
5	00100 01**000**	5
3	100**10 01101**	6
4	010**11 10110**	5

交叉

染色体	適合度	順位
10101 11**0**01	5	3
11100 10011	5	4
10101 11000	4	6
00100 01011	7	1
10011 10110	5	5
01010 01101	6	2

平均適応度　5.3

突然変異

第２世代

順位	染色体	適合度
3	10101 11**1**01	6
4	11100 10011	5
6	10101 11000	4
1	00100 01011	7
5	10011 10110	5
2	01010 01101	6

平均適応度　5.5

並び替え

染色体	適合度	順位
00100 01011	7	1
01010 01101	6	2
10101 11101	6	3
11100 10011	5	4
10011 10110	5	5
~~10101 11000~~	~~4~~	~~6~~

淘汰

平均適応度　5.5

順位	染色体	適合度
1	00100 01011	7
2	00100 01011	7
3	01010 01101	6
4	10101 11101	6
5	11100 10011	5
6	10011 10110	5

増殖

平均適応度　6.0

並び替え

染色体	適合度	順位
00100 0**1011**	7	1
10011 1**0110**	5	5
00**100 01011**	7	2
11**100 10011**	5	4
01010 **01101**	6	2
10101 **11101**	6	3

交叉

第３世代

順位	染色体	適合度
1	00100 00110	7
4	10011 11011	5
2	00100 10011	7
5	11100 01011	5
3	01010 11101	7
6	10101 01101	5

平均適応度　6.0

世代交代を続けていくと、最適解 00000 11111 へ近づくことが多い。（最適解が得られる保証はない。）

遺伝的アルゴリズムと遺伝的オペレータ

遺伝子の表現

一般に、GAにおいて、固体を表す文字列として、{0,1} の並びである文字列、すなわちビット列によるコード化が用いられる。ここで、各遺伝子は 0 または 1 の各ビットに対応している。また、文字列の長さも固定長とするのが一般的である。

[例] $f(x) = x \cdot \sin(10\pi x)+2.0$ の区間[-1,2]での最大値を求める

遺伝子：小数第６位の実数に対応させる。 $(-1.000000,\ 2.000000)_2$

→約$3 \cdot 10^6$の個体で22ビットの染色体で表現

適合度とスケーリング

適合度： 自然界では環境に対する適合尾の高い個体が生き残り、増殖する。逆に、適合度の低い個体は淘汰され、個体群の中から消滅することになる。このような適合度は一般に非負の数で表現されるので、GAにおいても、染色体 $S_i=\langle s_{i1}, s_{i2}, \ldots, s_{in}\rangle$ に対する適合度関数 $f(si)$は、$f(si)>0$が要求される。したがって、GAにおいては、対象とする最適化問題が、目的関数$z(x)$の最小化問題なのか、あるいは、目的関数 $u(x)$ の最大化問題なのかに対応して、それぞれの目的関数に対する次のような変換が一般的によく用いられる。

費用関数などの最小化問題の場合は、$C_{max} - z(x) \geqq 0$となるような、適当なC_{max}を導入して、適合度関数を以下のように定義する。

$$f(s_i) = (C_{max} - z(x_i)) \vee 0,\ \text{ただし、}s_i\text{はに}x_i\text{対応する染色体}$$

逆に、効用関数などの最大化の目的関数$u(x)$に対しても、$u(x) + C_{min} \geqq 0$となるような、適当なC_{min}を導入して、適合度関数を以下のように定義する。

$$f(s_i) = (u(xi)+C_{min}) \vee 0,\ \text{ただし、}s_i\text{はに}x_i\text{対応する染色体}$$

スケーリング： 各個体の染色体 $S_i=\langle s_{i1}, s_{i2}, \ldots, s_{in}\rangle$ に対する適合度関数$f(s_i)$が決定されたとき、この適合度関数による評価値をそのまま選択時の確率に反映させるよりはむしろ、何らなのフィルタを通して、評価値の差異を拡大あるいは縮小させるほうがより効果的な選択が行われる状況が考えられる。適合度のスケーリングとして、

(1)線形スケーリング
(2)シグマ切断
(3)べき乗スケーリング

などが提案されている。

(1) 線形スケーリング（のアルゴリズム）

手順１：個体群の平均適合度 f_{mean}，最大適合度f_{max}，最小適合度f_{min}を求め、c=2とする。

手順２：$f_{min} > (c \cdot f_{mean}-f_{max})/(c-1)$ならば、手順３へ、そうでなければ手順４へ進む。

手順３：$a=\{(c-1) \cdot f_{mean}\}/(f_{max}-f_{mean})$，$b=\{f_{mean} \cdot (f_{max}-c \cdot f_{mean})\}/(f_{max}-f_{mean})$として手順５へ進む

手順４：$a=f_{mean}/(f_{mean}-f_{min})$，$b=-(f_{mean} \cdot f_{min})/(f_{mean}-f_{min})$として手順５へ進む。

手順５：$i=1,2,..N$に対して、$f'_i=a \cdot f_i+b$なる線形スケーリングを行う。

(2) シグマ切断：線形スケーリングで$f'_i<0$となる場合が多いとき、すなわち個体群の多くの個体が高い適合度を持ち、小数の個体が非常に小さい適合度を持つとき、個体群の適合度の標準偏差σ切断が有効とされている。シグマ切断は、

$$f'_i=f_i-(f_{mean}-c \cdot \sigma),\ 1<c<3$$

(3) べき乗スケーリング

$$f'_i=(f_i)^k$$

遺伝的オペレータ

[選択]

(1) ルーレット選択：ルーレット選択とは、個体群の中の各個体の適合度の大きさの比例した選択確率によって選択する方法。

個体iが選択される確率p_iは、$p_i=f(s_i)/\Sigma f(s_j)$によって与えられる。

(2) 期待値選択：期待値選択とは、個体群の中の各個体の適合度とその総計を計算し、適合度の総計に対する各個体の割合で個体数を調整する方法。

[例]

適合度	12	2	20	22	34	64	8	24	10	4
期待値（総個体数１０）	0.6	0.1	1	1.1	1.7	3.2	0.4	1.2	0.5	0.2
選択数	1	0	1	1	2	3	0	1	1	0

(3) ランキング選択：ランキング選択とは、個体群の中の各個体の適合度の大きさではなく、順位に応じて選択する個体数を調整する方法

(4) トーナメント選択：トーナメント選択とは、個体群の中から定められた個数の個体をランダムに選択して、その中で適合度の一番高い個体を次世代に残すという手続きを次世代に残したい数の個体が選択されるまで繰り返すという方法。

(5) エリート保存選択：エリート選択とは、個体群の中の適合度が最も高い個体を無条件で次世代に残す方法。

[交叉]

選択された個体間での染色体の組換えにより新しい個体を生成する交叉は、GAでは、最も重要な遺伝的オペレータである。このような交叉は、個体群の中から任意の２個体（親）をランダムに選択し、さらに、ランダムに選ばれた１点あるいは多点の交叉点で遺伝子を組み換えることによって、新たな２つの個体（子）を生成する操作である。

(1) 1点交叉：1点交叉では、親１、親２の文字列上で交叉点"|"をランダムに１箇所選び、交叉点の右側の親の部分文字列を交換して、子１、子２を生成する。

[例] 親１：110000001 ⇒ 子１：110000100
親２：101110100 子２：101110001

(2) 多点交叉：多点交叉では、親１、親２の文字列上で交叉点"|"をランダムに複数箇所選び、交叉点の間で交互に親の部分文字列を交換して、子１、子２を生成する。

[例] ２点交叉
親１：110000001 ⇒ 子１：111110001
親２：101110100 子２：100000100

３点交叉
親１：110000001 ⇒ 子１：111110000
親２：101110100 子２：100000101

(3) 一様交叉：一様交叉では、まず、{0,1} をランダムに発生させて n ビットのマスクパターンを作る。このマスクパターン上の１の遺伝子座には親１の遺伝子を、0の遺伝子座には親２の遺伝子を受け継ぐ子１とその逆の子２を生成する。

[例]
親１ ：110000001
親２ ：101110100 ⇒ 子１ ：100010001
マスク：101101101 子２ ：111000100

[突然変異]

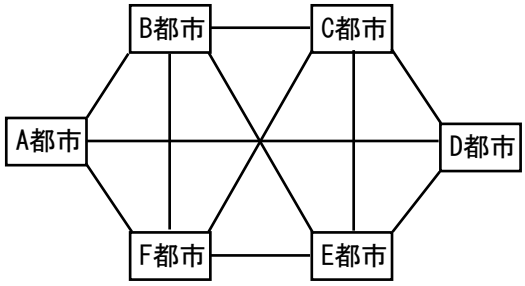
交叉だけでは、個体の親に依存するような限られた範囲の子しか生成できない。突然変異は、染色体上のある遺伝子座の値を他の対立遺伝子に置き換えることによって、交叉だけでは生成できない子を生成して、個体群の多様性を維持する働きをする。GAでは、各遺伝子座に対して定められた突然変異確率にしたがって対立遺伝子に置き換える。

[例]（第3遺伝子座に突然変異が起こる）

101**1**1101000 ⇒ 100**0**1101000

遺伝的アルゴリズムの応用例

巡回セールスマン問題への応用



ここで、染色体が訪問順序をそのままあらわしている場合、C都市→B都市→D都市→F都市→A都市→E都市の染色体は、CBDFAEとあらわされる。このような染色体の一点交叉を行う場合を考える。

親１：CBDFAE　⇒　子１：CBDFEC
親２：ADFBEC　⇒　子２：ADFBAE

すると、子１は、C都市を2回も訪問して、A都市を一度も訪問しないことになる。つまり、交叉によって解の満たすべき最低条件を満たさなくなっている。このような個体は致死遺伝子を持っているといい、淘汰される。このような致死遺伝子が発生すると無駄が多くなるので、コーディングおよび交叉の方法に工夫が必要。

[コーディングによる解決策]

順序表現

C→B→D→F→A→E のコード化

C → 3	A B C D E F 1 2 3 4 5 6
B → 2	A B D E F 1 2 3 4 5
D → 2	A D E F 1 2 3 4
F → 3	A E F 1 2 3
A → 1	A E 1 2
E → 1	E 1

染色体：322311

A→D→F→B→E→C のコード化

A → 1	A B C D E F 1 2 3 4 5 6
D → 3	B C D E F 1 2 3 4 5
F → 4	B C E F 1 2 3 4
B → 1	B C E 1 2 3
E → 2	C E 1 2
C → 1	C 1

染色体：134121

順序表現によるコーディングを用いて、1点交叉を行うと、

親１：322311 → 子１：322321 →　C→B→D→F→E→A
親２：134121 → 子２：134111 →　A→D→F→B→C→E

⇒解の満たすべき最低条件は満足する。

[交叉法による解決策]

1）部分一致交叉

１）２つの染色体において交叉させる部分を決める。
２）対応する文字の組を決める。
３）２つの染色体において、対応する文字同士を交換する。

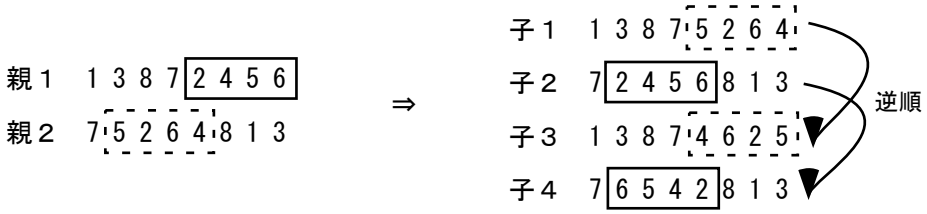
親１：CBDFAE
　　　||| (D⇔F, F⇔B, E⇔C)⇒ CBFDAE ⇒ CFBDAE ⇒ 子１：EFBDAC
親２：ADFBEC　　AFDBEC　　ABDFEC　　子２：ABDFCE

⇒解の満たすべき最低条件は満足する。

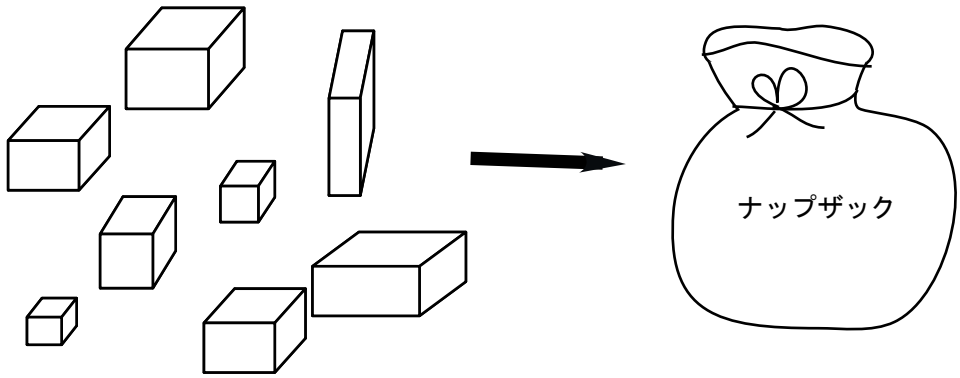
2) サブツアー交換交叉

サブツアー（部分的な訪問順序）を壊さないように交叉を行うアイデア

サブツアーを壊さないように、交叉の際、２つの親で共通の要素を含んだ部分を見つける。これらを交換あるいは逆順にすることによって子を生成する。



ナップザック問題への応用



ナップザック問題：ナップザックに荷物を入れる際、制限容量を越えない範囲でできるだけ詰め込む方法を求める（詰め込む荷物を決定する）問題。

応用例　１）一定の予算内での物資の購入
２）従業員の能力に基づいた人事管理
３）金融取引意思決定（ポートフォリオ選択）
４）貨物輸送システム

① 遺伝子型の決定

染色体の各ビットを、各荷物の「入れる」「入れない」に対応させる。例えば、荷物が10個あれば、10ビットの染色体になる。

② 適応度の評価

適応度を計算するとき、

個体が条件を満たしている　→　詰め込む荷物の容量の合計
個体が条件を満たさない　→　容量オーバー分に基づいた悪い値

③ 選択、交叉、突然変異

通常の方法で十分

50個の荷物に関するナップザック問題では、総あたりで求めるのと比較して数万倍速く計算できたという報告例がある。

まとめ

遺伝的アルゴリズムの特徴

多くの組み合わせの中からの選択能力に優れている。

評価関数の微分が不要。

遺伝的アルゴリズムの応用分野

設計問題、スケジューリング問題、組み合わせ最適化問題、制御問題、プログラムの自動生成（進化的プログラミング）

遺伝的アルゴリズムの採用理由

通常のアルゴリズムでは、現実的に求解が困難

評価関数が微分できない。

遺伝的アルゴリズムの応用のチェックポイント

次の点を考慮した上で、利用を考える。

１）問題が遺伝的アルゴリズムによる解法に適しているか。

２）解の明確な評価法があるか（適応度を明確に与えられるか）。

３）解（解の候補）を遺伝子・染色体として表現できるか。

４）実際のシステムで解が求まらなかった場合の対応策