

---



---

## 第1回 ECゼミ

---



---

ゼミ担当者 : 田中 裕也, 荒久田 博士, 昌山 智  
 指導院生 : 青井 桂子, 森 隆史, 米田 真純  
 開催日 : 2003 年 4 月 18 日

ゼミ内容: 知的システムデザイン研究室の研究テーマの1つである, Emergent Computation(EC), また EC における Genetic Algorithm(GA), Simulated Annealing(SA) についての理解を深めることを目的とする.

### 1 はじめに

知的システムデザイン研究室 (以下, ISDL) の目的は「知的なシステムを設計する」ことである. ISDL の研究の柱は,

- 知的化・コラボレーション
- 最適化
- 並列・クラスタ

の3つであり, これらの3つのテーマは, それぞれ密接に関係している. 最適化という手法を用いて, より良いシステム的设计を行う. ISDL では, 最適化を行う際に確率的手法を用いている. 確率的手法では, 多くの反復計算が必要となる. そこで, 現実的な時間で解くためにもスピードアップは必須であり, 並列化による高速化をはかる必要がある. そして, 知的化・コラボレーション班では知的なシステムを作るためのツールとして最適化, 並列化を行っている.

ECゼミでは, ISDL の研究の柱の1つである最適化の内, 創発的計算について説明する. 最適化ゼミでは, 制約条件や古典的手法などの最適化の研究の本質的な部分を説明していたが, ECゼミでは SA, GA のような, 現在 ISDL で主に使っている創発的な最適化手法を説明する.

## 2 EC

### 2.1 EC とは何か?

EC(Emergent Computation: 創発的計算) は最適化の手法の一種である. 最適化の手法としては, 大きく2つに分けることができる. 1つは「最急降下法」や「欲張り法」といった古典的手法であり, これらの手法は, 解ける問題が少ないため, 実問題では使えないことが多い. もう1つは, 今回紹介する EC である. EC は, 計算量は膨大だが, 広範囲な問題にも対応できるのが特徴である. つまり, 計算パワーに任せて問題を解くというような解決法である.

### 2.2 EC の特徴

- 長所
  - 比較的短い時間で最適解に近い解が求まる.
  - 解法が確立されていない問題にも有効である. また, ブラインドサーチ<sup>1</sup>も可能である.
- 短所
  - 常に最適解が得られるとは限らない.
  - 計算量が膨大になり, コンピュータの高速化が要求される.

### 2.3 EC の手法

EC(創発的計算) としてどのような手法があるのか以下に述べる.

- Evolutionary Computing  
 Evolutionary Computing(進化的計算) とは, 生物進化を計算機上で模倣し, 適合, 学習, 最適化などの機能を実現しようという手法である. 進化型計算の研究は, いずれも 1960 年代に独立して始まった以下の3つのグループの活動を起源としている.
  - 遺伝的アルゴリズム  
 (Genetic Algorithm: GA)  
 問題の解を記号列として表現した個体を用い, 生物の遺伝的機構と自然選択を模倣して, 個体集合による離散的な探索空間での適応や最適化を扱う.

<sup>1</sup>目的関数の形は未知であるが, 設計変数の値から, その目的関数の値が求められるもの

- 進化プログラム (Evolutionary Programming : EP)  
個体集合と選択を用いることについては GA と共通しているが、新しい探索点の生成には専ら突然変異を用いる。
- 進化戦略 (Evolutionary Strategy : ES)  
実数関数の最適化を対象とし、個体表現には実数値をそのまま用いる。EP と同様、個体集合の選択と突然変異を主とする探索手法である。

- SA (Simulated Annealing)  
SA は、最適化問題、特に組み合わせ最適化問題<sup>2</sup>のための汎用的近似解法の 1 つである。

### 3 Simulated Annealing(SA)

#### 3.1 SA の概要

シミュレーテッドアニーリング (Simulated Annealing: SA) は Kirkpatrick らによって提案された組み合わせ最適化問題のための近似解法の 1 つである。組み合わせ最適化問題とは、最適化問題の中でも、設計変数値が整数などの集合、あるいは、数列で与えられている場合の問題のことを言う。システムの計画や運用などの効率化を考える場合、多くの問題が組み合わせ最適化問題として定式化できるが、実際的な問題の多くは厳密な最適解を求めるのは困難である。そこで、この種の問題は満足できる解を求める近似解法が適用されている。SA は、Fig. 1 のように高温で溶融状態にある金属を徐々に冷やすことによって、もとの金属より欠陥の少ない優れた結晶構造を作る物理プロセス (焼きなまし) を計算機上に模倣した手法である。

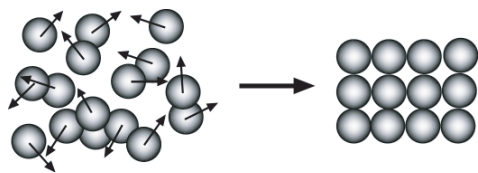


Fig. 1 原子の動き

SA の様子を模式的に示したものが、Fig. 2 である。SA は、次の状態  $x'$  が現在の状態  $x$  よりも良い目的関数値を与えるならば、暫定解を  $x'$  に推移する。また、次の状態が現在の状態よりも悪い目的関数値となる改悪方向への推移も一定の制限のもとで許す。このため、目的関数が複数の極小値を持つ Fig. 2 のような場合でも、最適解  $x^*$  に到達することができる。それに対して、局所探索法は解の改善操作を繰り返す最適化手法である。このため、Fig. 3 のように目的関数が複数の最小値を持つ

<sup>2</sup>具体的には巡回セールスマン問題など

場合、初期値を  $x^0$  に与えて探索を行うと、近傍内に改善する点が存在しない  $x^3$  の状態で停止し、最適解  $x^*$  には到達できない。

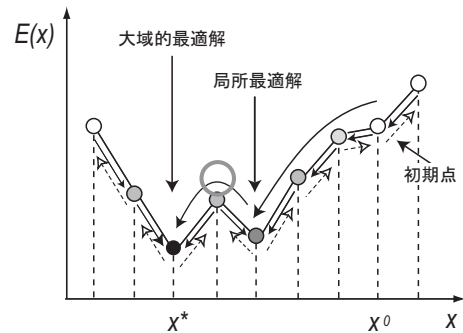


Fig. 2 シミュレーテッドアニーリング

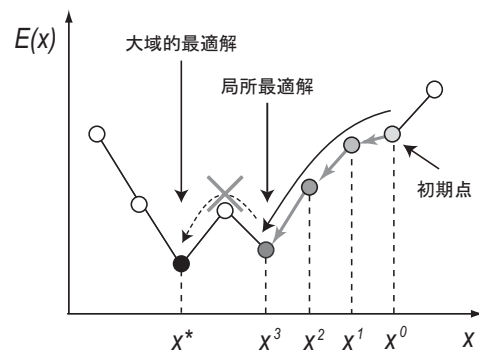


Fig. 3 局所探索法

#### 3.2 SA のアルゴリズム

SA は、与えられた初期状態から出発して、エネルギーが確率的に小さくなるように次々と状態を変化させ、最終的には最適状態になることが期待されるアルゴリズムである。

##### 3.2.1 SA の基本となるアルゴリズム

SA の基本アルゴリズムは生成、受理、クーリングから成り立つ。SA の基本的なアルゴリズムを Fig. 4 に示す。

##### 1. 初期設定

- 温度  $T$  を初期化する ( $T_k = T_1$ ) 。
- 初期状態  $x_0$  を与え、初期状態でのエネルギー  $E$  を計算する。

##### 2. 現在の温度 $T_k$ で一定期間、次の処理を繰り返す。

- 現在の状態から次の状態  $x'$  を生成する。
- 次の状態  $x'$  でのエネルギー  $E'$  を計算する。
- エネルギーの差分  $\Delta E (= E' - E)$  と温度  $T_k$  を用いて、次の状態を受理するか否かの判定を行う。

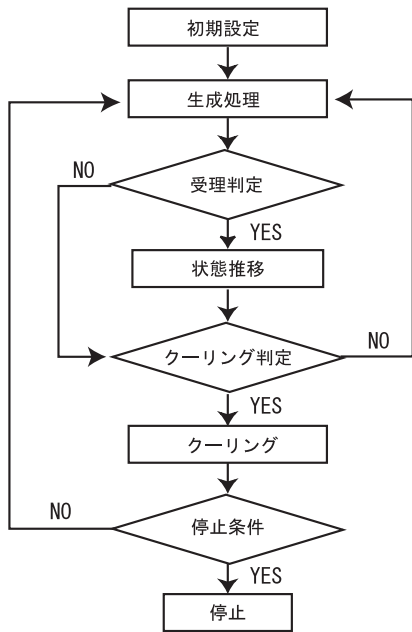


Fig. 4 SA のアルゴリズム

- 受理する場合は次の状態に推移する ( $x'$  が  $x$  に,  $E'$  が  $E$  となる) .

以上の処理をアニーリングと言う .

### 3. クーリング

- 一定期間アニーリングを行った後にクーリングを行い, 次の温度  $T_{k+1}$  を求める .
- 再びアニーリングを行う .

### 4. 終了

- 温度が十分に下がり, 停止条件に達すればそのときの  $x$  を最適状態,  $E$  を最適値として終了する .

#### 3.2.2 生成処理

生成処理では, 現在の状態  $x$  が与えられ, 次に推移すべき状態  $x'$  を求める . この処理では, 状態  $x$  から出発して状態  $x'$  が生起する確率分布  $G(x, x')$  を用いる . 組み合わせ最適化問題の場合は, 状態  $x'$  は状態  $x$  の近傍にあたり, 推移に優先性を与えない場合は式 (1) のように等確率推移となる .

$$G(x, x') = \frac{1}{n(x)} \quad (1)$$

#### 3.2.3 受理判定

受理判定は, 次の状態  $x'$  のエネルギー  $E'$  と現在の状態  $x$  のエネルギー差  $\Delta E (= E' - E)$ , および温度パラメータ  $T$  によって, 次の状態への推移を受理するか

否かの判定を行う . 通常は式 (2) の Metropolis 基準が採用される .

$$ACCEPT(E, E', T) = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp(-\frac{\Delta E}{T}) & \text{otherwise} \end{cases} \quad (2)$$

この基準が返す値は受理する確率で, '1' なら次状態に推移し, '0' なら推移しない . この値は温度  $T$  によって変化し, Fig.5 を見てもわかるように, 温度  $T$  は, エネルギーが増大する方向への推移確率に重大な影響を与えるパラメータである . このように温度が高い場合は悪い状態への推移確率も大きくなり, 反対に温度が低い場合は良い方向に推移することになる . しかし, どんな温度でも悪い方向への推移確率が '0' になるわけではない .

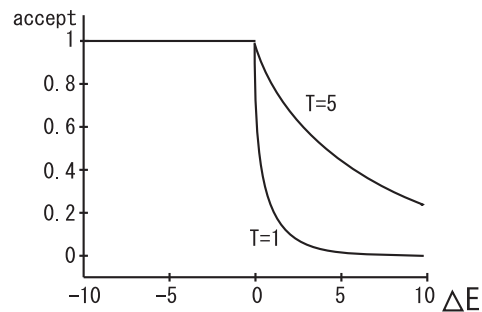


Fig. 5 状態遷移の受理確率

#### 3.2.4 クーリング

クーリングでは, 最適解の漸近収束性を保証するためには式 (3) に示す対数型アニーリング以上に急速に冷やしてはいけない .

$$T_k = \frac{T_1}{\log k} (k \geq 2) \quad (3)$$

しかし, 式 (3) は現実の応用ではあまりにも計算スピードが遅いため, 運用上は真の最適解への収束を犠牲にして式 (4) に示す指数型アニーリングが用いられる .

$$T_{k+1} = \gamma T_k (0.8 \leq \gamma < 1) \quad (4)$$

クーリングは, 各温度で状態  $x$  が平衡状態が実現するまで "十分な探索" を行った後に実行する . この "十分な探索" としては

1. 定めた回数だけ繰り返す .
2. 定めた受理 (または棄却) 回数になるまで繰り返す .
3. 温度幅を十分小さくする場合は 1 回受理するまで繰り返す .

などの実装方法がある . どの方法が良いかはそれぞれの実験で決める .

### 3.2.5 停止条件

SA の処理が停止するための条件には、次のような方法がある。

1. 定めた回数  $k_f$  だけアニーリングを繰り返して停止
2. 受理がほとんど起こらなくなったら停止
3. 同じ状態が何度も生成されるようになったら停止
4. 温度が十分低くなったら停止
5. エネルギーの変化またはエネルギーそのものが十分小さくなったら停止

### 3.3 SA の特徴

SA の特徴としては、以下のものが挙げられる。

#### 3.3.1 長所

- 多くの最適化手法が局所最適解に陥ってしまうという欠点を持っているのに対して、SA は容易に局所最適解に陥らず、理論的には真の最適解が得られることが証明されている。これは解が改良される方向のみだけでなく、改悪の方向にも探索が進むという仕組みによるものである (Fig. 2)。
- アルゴリズムが極めて汎用に出来ているため、広範囲の問題に適用することができる。
- 目的関数に関する制約がほとんどなく柔軟である。簡単に言うと、目的関数は微分可能でなくても、複雑な式で求まるものであっても、確率的であってもよい。
- 実装が容易である。

#### 3.3.2 短所

- 最適解を求めるためには長い計算時間が必要である。そのため、逐次処理のまま高速なアニーリングを導入する高速化の研究、および並列化して高速化を図る並列化の研究が行われている。
- 汎用解法であるために、問題を解くために必要なパラメータチューニングなどを個別に行う必要がある。特に温度を制御するパラメータのチューニングは非常に困難である。

### 3.4 ISDL における SA の研究

ISDL における SA の研究を以下に示す。

- タンパク質のエネルギー最小化による立体構造予測
- TSP (巡回セールスマン問題)
- JSP (ジョブショップスケジューリング問題)

## 4 Genetic Algorithm(GA)

### 4.1 GA の歴史

GA は、1960 年代の終わりから 1970 年代のはじめに、J.Holland と彼の同僚やミシガン大学の学生らによって、自然界のシステムの適応過程を説明し、生物進化のメカニズムを模擬する人工モデルとして提唱されてきた手法であると見なされている。その後、応用面での実用的な成果はあるものの、理論的な研究の発展がそれほど見られなかったため、比較的最近になるまで世間からはそれほど注目されなかった。しかし 1985 年に、カーネギー・メロン大学において、第 1 回の遺伝的アルゴリズムに関する国際会議が開催されたことが契機となり、最適化・適応・学習のための方法論として急速に注目されはじめ、多方面に適用されるようになってきている。

### 4.2 GA の概要とプロセス

#### 4.2.1 GA の概要

遺伝的アルゴリズム (Genetic Algorithm: GA) は、適応範囲の非常に広い、生物の進化を工学的にモデル化し、また参考にした学習的アルゴリズムである。基本的な GA として、Goldberg によって提案された単純 GA (simple genetic algorithm: sGA) があり、そこから種々な手法が提案されている<sup>3</sup>。

GA では、ある世代 (generation) を形成しているいくつかの個体 (individual) の集合を母集団 (population) と呼ぶ。また、GA では自然界における生物の進化過程と同様に、環境への適合度 (fitness) の高い個体が高い確率で選択 (selection) される。その個体に対して、遺伝子の交叉 (crossover) や突然変異 (mutation) が、ある確率で発生することによって次の世代の個体群が形成されてゆき、最後に得られた個体群の中で最も適合度の高い個体が最良システムとなる。

各個体は、それぞれ染色体 (chromosome) によって特徴付けられており、染色体は遺伝子 (gene) の集まりで構成されている。また、通常 GA では 1 染色体で 1 個体を表現 する。染色体上で各遺伝子のおかれている位置を遺伝子座 (locus) といい、対立遺伝子 (allele) としては、2 進数のビット {0, 1} を用いることが多い。GA で扱う情報は遺伝子型 (genotype: Gtype) と表現型 (phenotype: Ptype) の二層構造からなる。

- 遺伝子型...GA のオペレータの操作対象、染色体の構造にあたる (ex. 01110, 00110; 0, 1 のビット列)。
- 表現型...環境に応じて表現型から適合度が決まる、環境内での構造を示す (ex. 9, 5; 具体的な値)。

なお、表現型から遺伝子型へ写像することをコード

<sup>3</sup>分散 GA(DGA), 環境分散 GA(DEGA) など

化<sup>4</sup>(coding), 逆に遺伝子型から表現型へ逆写像することをデコード化(decoding)という。Ptypeによって適合度が決まり, その適合度が大きいものほど子孫を作りやすく, 小さいものほど死滅しやすい(淘汰)ようになっている。このことにより, 次世代の各個体の適合度は前世代よりも良いことが期待される。

#### 4.2.2 GAのプロセス

1. 問題のモデル化: 対象問題から, 染色体の遺伝子配列・適合度関数・GAパラメータの数値を決定する。
2. 初期母集団の生成: 初期母集団を構成する個体を生成する。
3. 染色体の評価: その環境における各個体ごとの適合度を算出する。
4. 適合度での選択: 各個体の適合度に基づいて次世代個体を決定する操作。
5. 交叉処理: 複数の個体内の染色体の一部を組み替え, 新しい個体を生成する。
6. 突然変異操作: 個体上のある遺伝子を一定の確率で変化させる。
7. 終了判定: 終了条件を満たしていれば, 母集団内の適合度が最も高い染色体を解とする。

このようなGAが古典的な探索法と大きく異なる点は, Goldbergによれば, 次の4つの特徴にある。

- 設計変数を直接操作せずにコーディングしたものを扱う。
- 一点探索ではなく, 多点探索である。
- サンプルングによる探索で, ブラインドサーチである。
- 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。

#### 4.3 GAの特徴

##### 4.3.1 GAの長所

- 任意の解の良さ(適合度)を一気に決められる問題であれば適応できる(適応できる問題が広い)。
- 組み合わせ爆発を起こすような広大な探索空間を持つ問題に対しても適用できる。
- 局所解をもつ問題に対しても有効である。

##### 4.3.2 GAの短所

- 広域な探索は得意だが, 局所探索は不得手である。
- より良い解を求めるには個体数を増やす必要があるが, 個体数を増やすと計算に時間がかかる。

<sup>4</sup>具体的には, binary-coding, Gray-codingなどの手法がある。

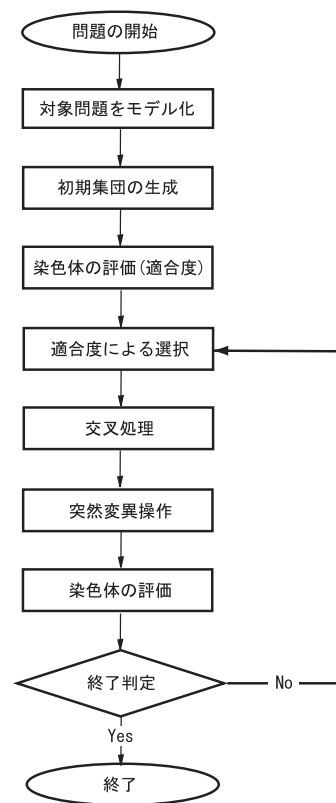


Fig. 6 GAの構成プロセス

- 早熟収束<sup>5</sup>によって, 局所解へ収束してしまう。
- パラメータの設定が複雑である。

#### 4.4 GAの例… OneMax問題

OneMax問題とは染色体内においてすべての遺伝子が1となることを最適解とする問題のことである。これにより, 1のビットを多く持つ個体を高く評価する事にする。

Fig. 7にOneMax問題におけるGAの探索の様子を示す。

1. 問題のモデル化: 遺伝子長5の染色体を4つ生成する。
2. 初期母集団の生成: 1で生成した染色体の遺伝子にランダムな数字(0, 1)を入れる。
3. 染色体の評価: OneMax問題の場合は1が多くある染色体ほど評価を高くする。例えば  $11111$  は4,  $11110$  は3,  $11101$  は2,  $11011$  は1と評価する。
4. 適合度での選択:  $11111 > 11110 > 11101$  の順に適合度が高いので, 選択もこの順で選択されやすくする。Fig. 7の場合,  $11111$ ,  $11110$ ,  $11101$  が選択されている。
5. 交叉処理: 複数の親(一般には2つ)の持つ染色体を交換し, 別のより良い遺伝子を受け継ぐ新し

<sup>5</sup>補足資料のLINGOを参照

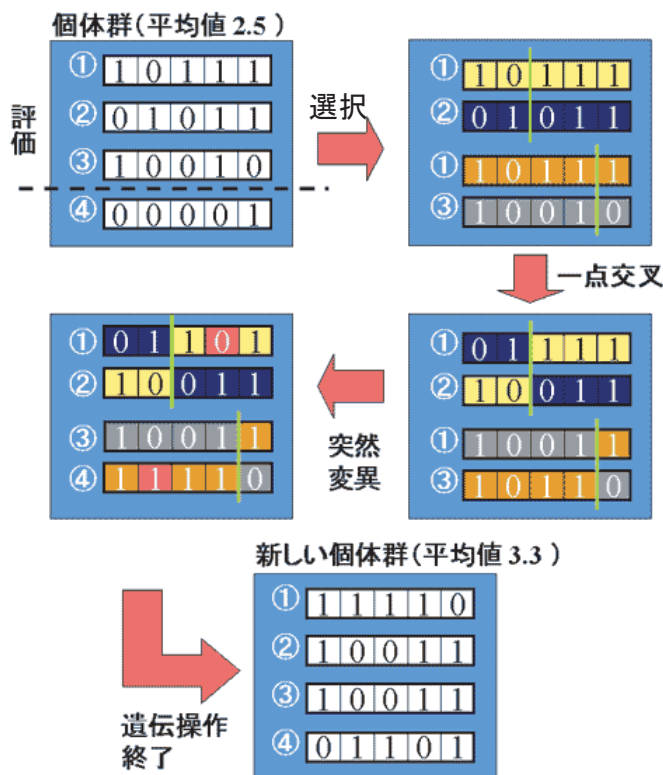


Fig. 7 GA による探索の様子

い個体を生成する．どの場所で交叉させるかはランダムであり， と は 2 つ目の区切りで， と は 4 つめの区切りでそれぞれ遺伝子を交換している．

6. 突然変異操作：あらかじめ定められた一定の割合で，選ばれた個体の遺伝子を変化させる．Fig. 7 の場合， の 4 番目の遺伝子と の 2 番目の遺伝子に変化している．
7. 終了判定：2 から 6 を終了条件を満たすまで繰り返す．終了条件を満たせば終了．

#### 4.4.1 ISDL における GA の研究

ISDL における GA の研究を以下に示す．

- ディーゼルエンジンの燃料噴射スケジュールの最適化
- ゴルフボールの回転角検出
- トラス構造物最適化
- JSP (ジョブショップスケジューリング問題)
- タンパク質のエネルギー最小化
- TSP (巡回セールスマン問題)

#### 参考文献

- 1) 伊庭齊志 “ 遺伝的アルゴリズムの基礎 ”，オーム社，1994．
- 2) 金子美華，渡邊真也 “ GA 基礎講座 (2) ”，1998．
- 3) 畠中一幸 “ 卒業論文 遺伝的アルゴリズムの分散並列化 ”，1998．
- 4) GA について，<http://www.kuroda.elec.keio.ac.jp/projects/TeamGA/chapter1.htm>

LINGO

世 代	... GA における手順の繰り返しの 1 単位
個 体	... 解に対応するもの
適 合 度	... 個体の良好さを示す値
染 色 体	... 個体を記号列や文字列で表したものの
遺 伝 子	... 染色体を構成する最小単位, GA の最小構成要素
遺 伝 子 座	... 各遺伝子のおかれている場所をさす
対立遺伝子	... 各遺伝子が行うことのできる遺伝子候補
遺 伝 子 型	... オペレータの操作対象となる染色体構造
表 現 型	... 環境内での構造を表す. 与えられた環境内での構造表現型によって適合度が決まる
コ ー ド 化	... 遺伝子型から表現型への変換. Binary-coding, Gray-coding などがある
局 所 解	... 最適解ではない, 極大 (極小) 値などの解
早 熟 収 束	... 探索序盤に他より極端に適合度の高い個体が存在した場合に, その個体が母集団内に急速に広がり, 母集団内の多様性が失われ, 局所解に陥ってしまう現象
選 択	... 適合度の高い個体を次世代に残すためのオペレータ. ルーレット選択・ランキング選択・トーナメント選択などがある
エ リ ー ト	... 母集団の中で最も適合度の高い個体. エリート戦略は次世代にエリート個体が必ず保存される
交 叉	... 複数の個体の染色体を組み替えるオペレータ. 一点交叉・多点交叉・一様交叉が代表的
交 叉 率	... 交叉を行う確率を定めたもの
突 然 変 異	... ある遺伝子を他の対立遺伝子に置き換えるオペレータ
突然変異率	... 突然変異を行う確率 $1/L$ (染色体長) が一般的
致死遺伝子	... 解の候補となり得ない遺伝子型の個体. GA における探索の妨げとなるため致死遺伝子の発生を防ぐ必要がある
多 様 性	... 母集団全体で各個体にどれだけ差があるかを示す