

# Keyword: Evolution (進化)

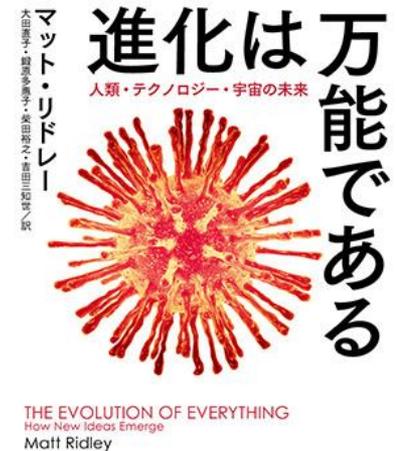
---

## ■ Research Topics

- Evolutionary computation (進化型計算手法)
- Evolutionary systems (進化型システム)
- Artificial life (人工生命)
- Genome informatics / Bio-informatics

# 進化をめぐる研究

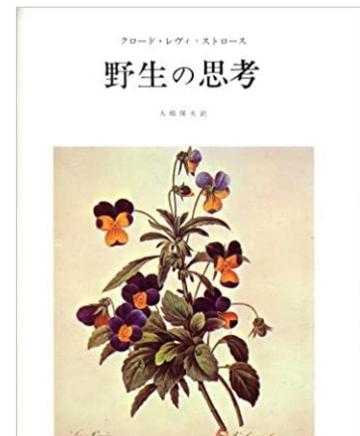
- 進化計算、複雑系、人工生命、合成生物学。。。
- 「進化は万能である」
- 「進化はわれわれよりも賢い」
  - フランシス・クリック
- 「創造とは組み換えである」
  - フランソワ・ジャコブ



# 進化をめぐる研究



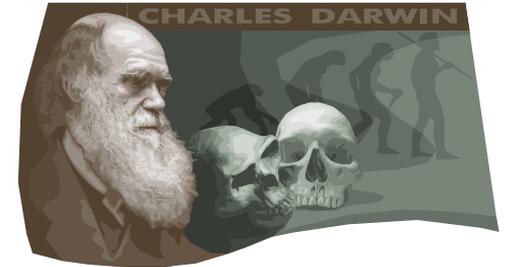
- クロード・レヴィ＝ストロース(1908-2009)
  - 「ブリコラージュ」
    - 余り物や端切れかを使い本来の目的と関係なく役立つ道具をつくること
    - 人類が古くからもつ知のあり方「普遍的な知性」
  - 近代以降のエンジニアリング思考＝「栽培された思考」



『野生の思考(1962)』

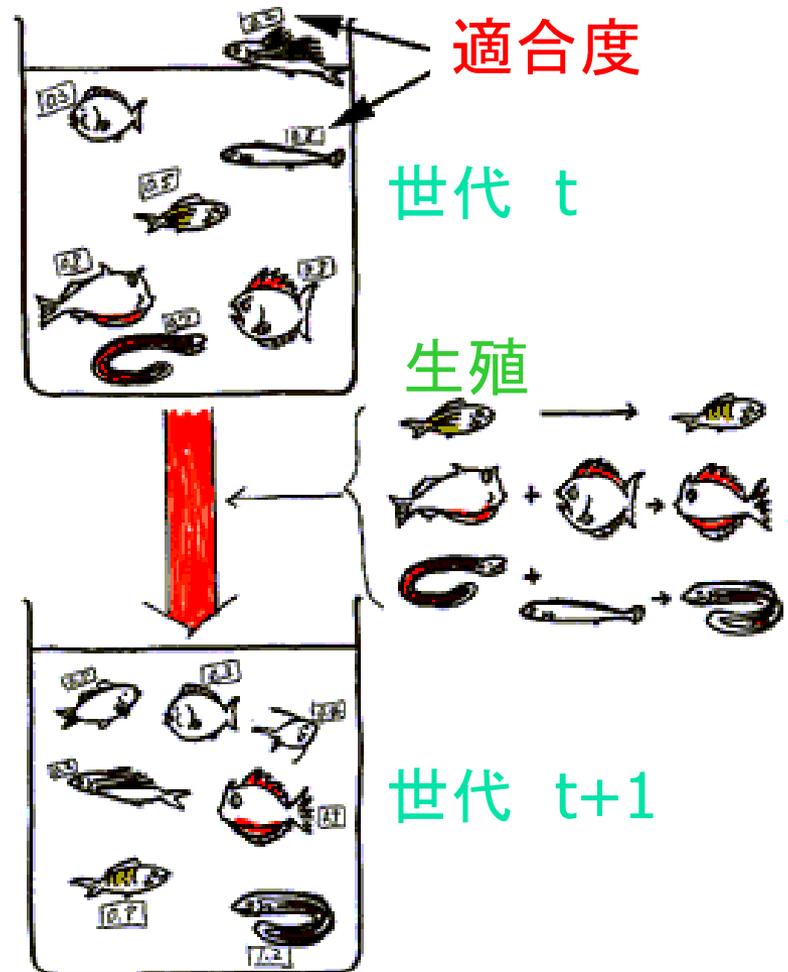
# ダーウィンの自然選択

- 「変異とは、事実上すべての動物や植物のグループがもつ特徴である」
- 「生物のグループは、すべて子孫を過剰に生み出している」
- 「もっとも環境に適応するものが生き残る」
  - 「もっとも適応的な特徴がどの程度でも遺伝すれば、この有利な形質は次の世代に伝えられる」

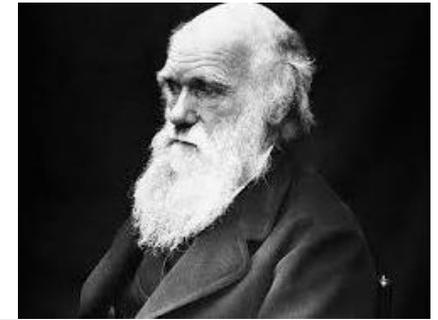


# 進化論的計算とは何か？

- Survival of the fittest
  - 適者生存
- Reproduction
  - 生殖
- Variation
  - 変容

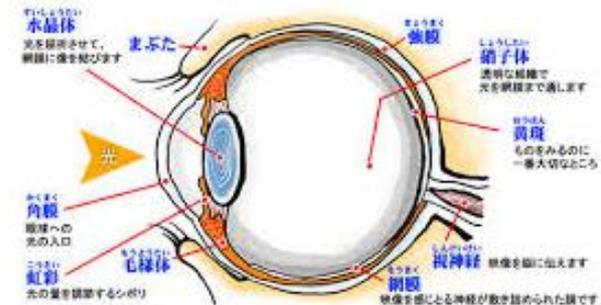


# 眼の進化



- ダーウィンを悩ませた問題
- 眼のような複雑な機能の進化は「**このうえなく不条理**のことに思われる」（種の起源）

今日に至るまで、眼は私を身震いさせる(ダーウィン)

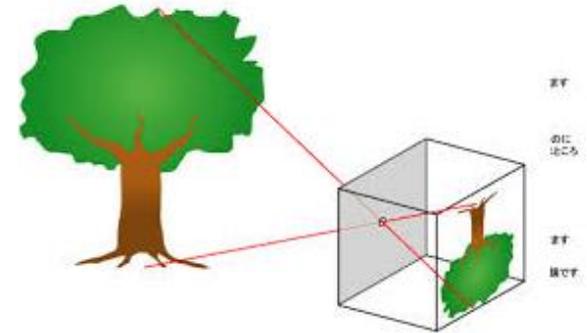


- この様な複雑な器官が**突然変異**で生じるのか？
- 最初の一步(眼的なもの)は**生き残れる**のか？

半分の眼は一体何の役にたつのか???

# 第一の疑問： 半分の眼は役に立つか

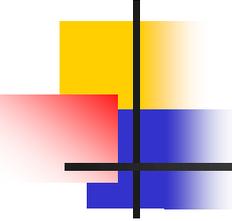
- 単純な眼でもないよりまし
- 走光性
  - シアノバクテリアが光に応答した運動性
- ピンホールカメラ
  - オウム貝
    - アンモナイトの親戚
    - 貝に棲むタコの仲間



ピンホールカメラ



@PNG (2005)  
教科書, p.115,  
紋様の創発シミュレーション



## 第二の疑問： 不可能な山を登る？

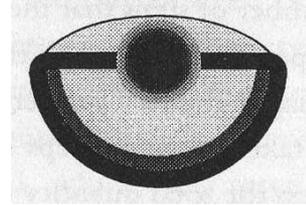
- 走光性、ピンホールカメラ
  - 突然変異で確かにできそう
- 「レンズ」はどうやって進化するのか？
  - 不可能な山
  - たくさんの小さな峰：オウム貝、シアノバクテリア

コンピュータシミュレーションによる説明が行われた

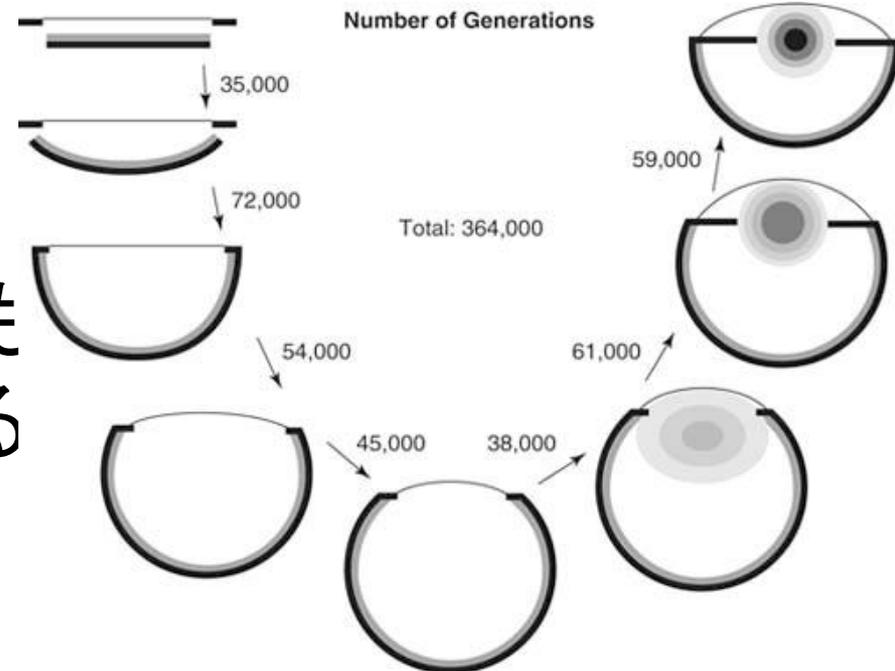
Nilsson, D.-E. and Pelger, S. (1994)

A Pessimistic Estimate of the Time Required for an Eye to Evolve. *Proceedings: Biological Sciences*, 256 (1345): 53-58.

# 眼の進化シミュレーション



- 濃度勾配型のレンズは第6段階で現れる
- 未発達な段階(平らな皮膚のよな構造)から出発して魚類の眼(レンズをもったカメラ眼)が進化するまでには40万世代もかからない



# What is Evolutionary Computation?

- Genetic Algorithms  
遺伝的アルゴリズム

- Genetic Programming

遺伝的プログラミング



E.g., N700系、カーナビ、エレベータ

# GTYPE & PTYPE

Genotype      Phenotype  
遺伝子型      表現型

PTYPE ← GTYPE



Mappingは一方のみ  
(セントラルドグマ)

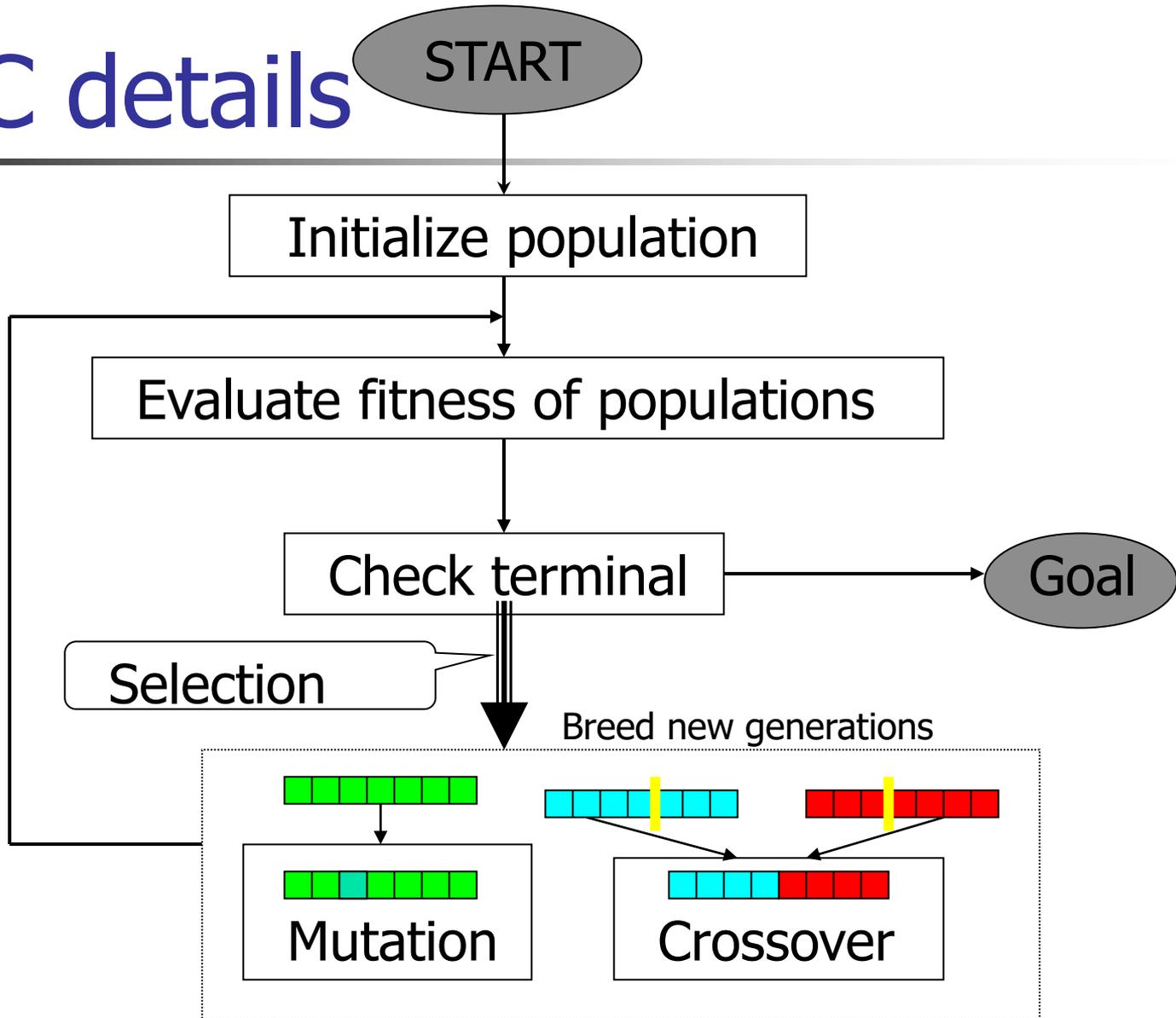


Fitness  
適合度

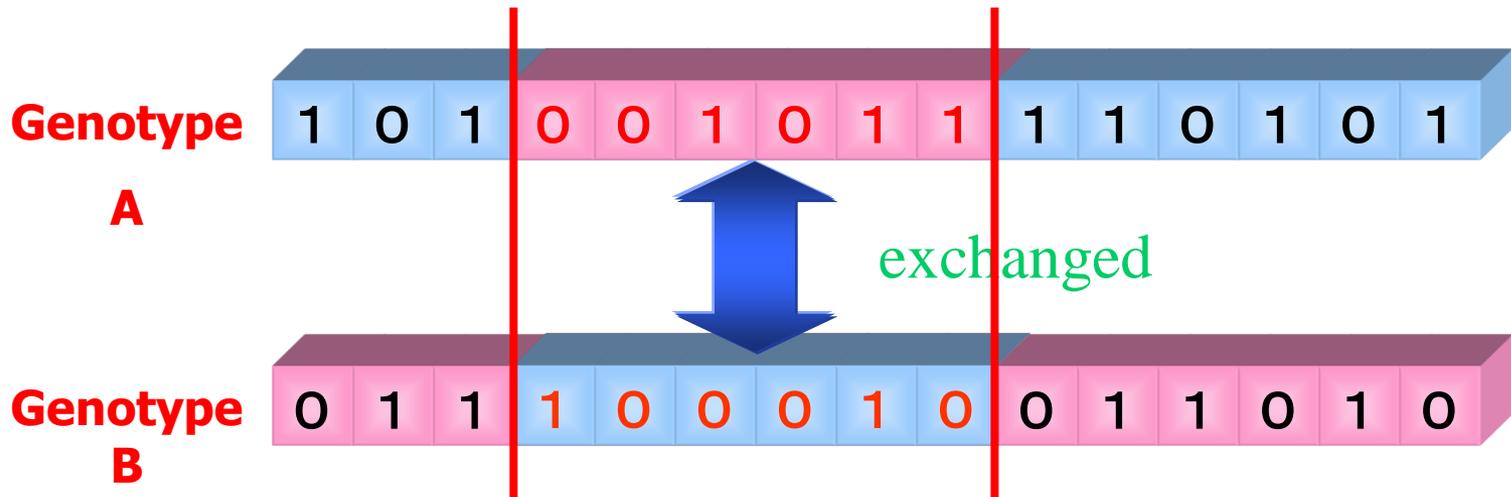


PTYPEにのみ作用する  
(GTYPEにはよらない)

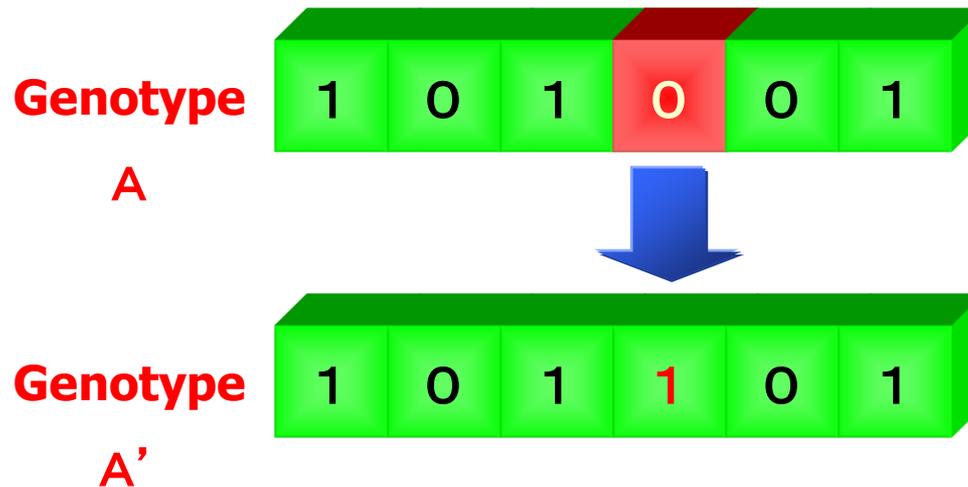
# EC details



# GA Operation: Crossover



# GA Operation: Mutation

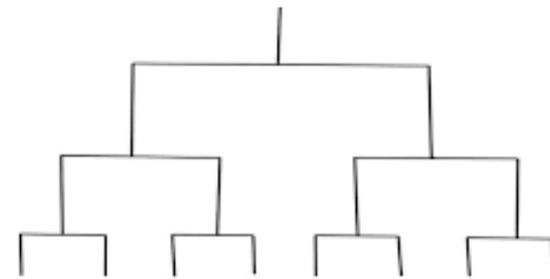


# 選択の方法



- ルーレット選択
  - 重み付きのルーレット

- トーナメント選択



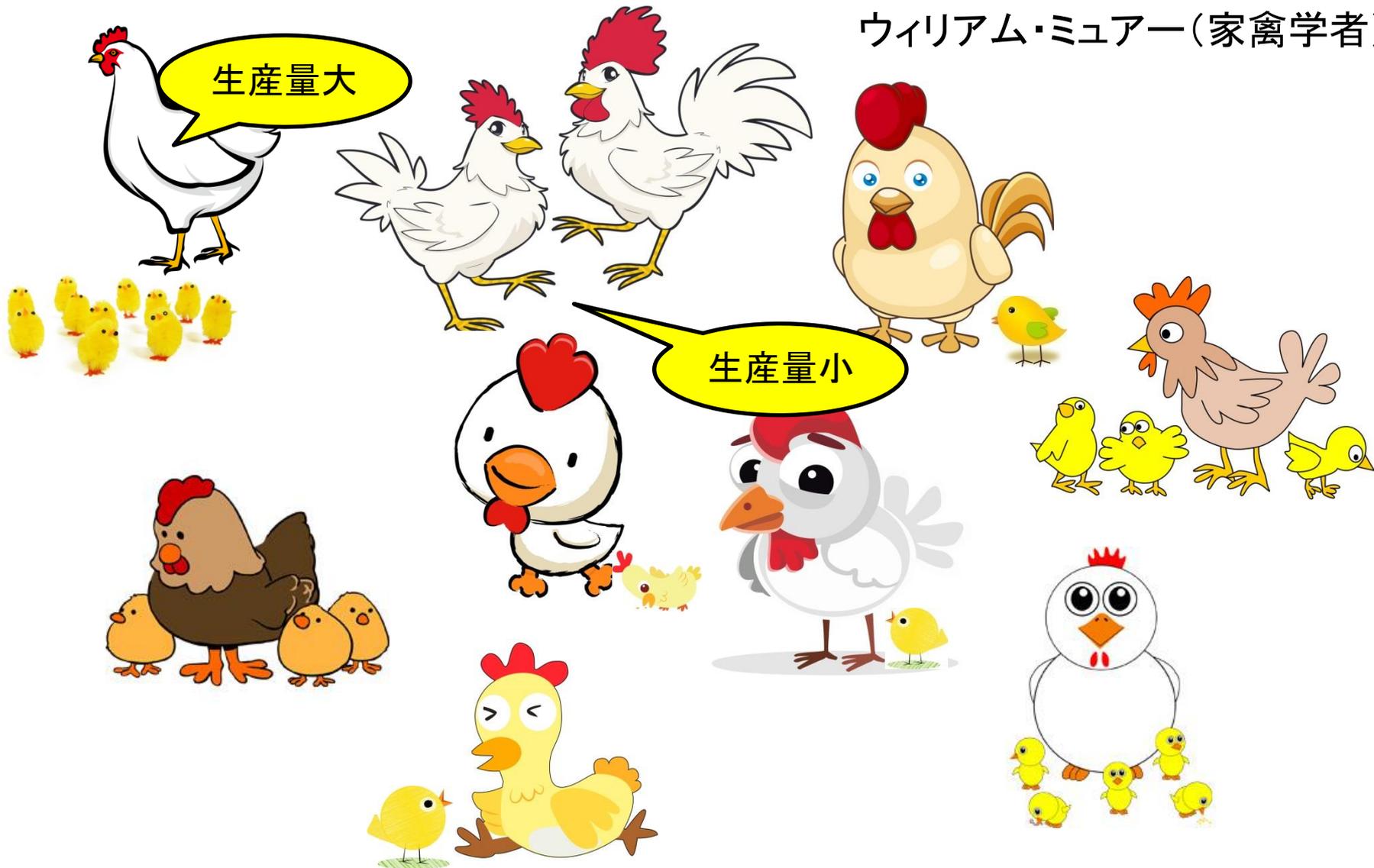
- エリート戦略
  - 成績トップの何%かを必ず残す
  - 環境が不変の場合、世代交代で成績が悪くならない
  - ただし、多様性は失われるので注意

# たくさん卵を産む鶏を得るには？

ウィリアム・ミュアー(家禽学者)

生産量大

生産量小



# たくさん卵を産む鶏を得るには？

- 現代的な養鶏産業
- 1つの檻＝九羽から十羽のメンドリ
- 選択的繁殖を行って卵の生産量を増やしたい
- 方法1：多数の檻から卵をいちばんよく産むメンドリを選び、次の世代のメンドリを繁殖させる。
- 方法2：産卵量が一番多い檻のメンドリを全部選択して次の世代のメンドリを繁殖させる。

# 何が悪かったか？

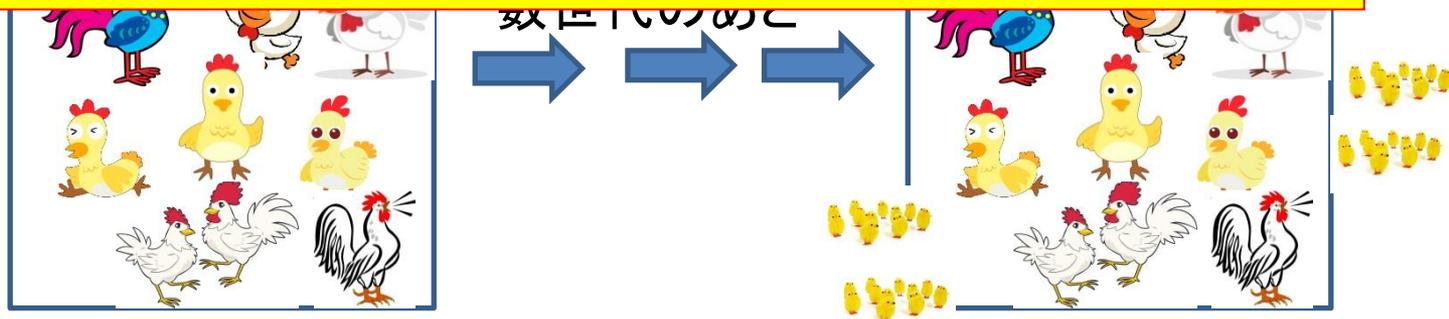
- 方法1: 多数の鑑から卵をいちばんよく産むメンドリを選び、次の世代のメンドリ繁殖させる。

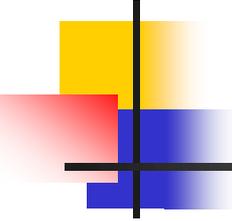


- いちばんたくさん卵を産むメンドリは、檻のほかのメンドリの産卵を抑制することで、いちばんたくさん卵を産むようになっていた。

方法2  
世代

- 第一の方法では、それぞれの檻で最もたちの悪いメンドリを選択したため、6世代後には情緒不安定になって破綻が生じた。





# GA Theory

---

## ■ Schema

- useful building blocks

- $111^*01 = \{111101, 111001\}$

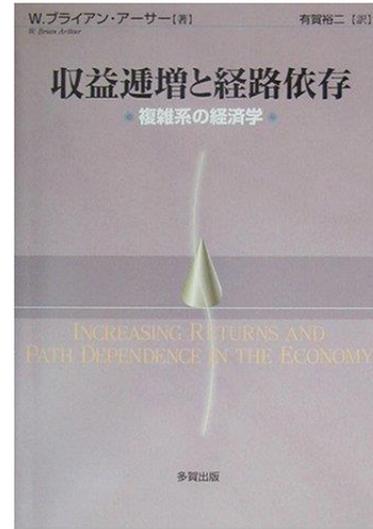
## ■ Schema Theorem

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left\{1 - p_c \cdot \frac{\delta(H)}{l-1}\right\}$$

# 2-armed bandit



- 2本のレバーがある**スロットマシン**に使うためにN枚のコインが与えられた
- それぞれのレバーA1,A2を一回引くことにそれぞれ**平均 $m_1, m_2$ 、標準偏差 $\sigma_1, \sigma_2$** で賞金が得られる。
- 2本のレバーから出てくる賞金は毎回独立で定常(時間により不変)とする
- あなたは $m_1, m_2, \sigma_1, \sigma_2$ の値は知らない。
- 目標はN回の試行で得られる**賞金総額を最大化**することである。
- このとき各レバーへの試行配分(残りのコインの賭け方)はどのようにすればいいか？



# 2-armed bandit

Exploration (調査)  $\mu_R > \mu_L$  なのか  $\mu_R < \mu_L$  なのかを決定する.

Exploitation (利用) 上の決定にしたがって支払い率の良いレバーに賭ける.

- UCB値を用いる(ゲームの探索:UCTアルゴリズム):割と良い

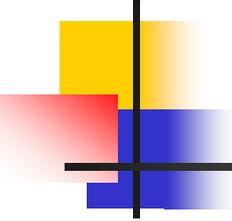
$$\text{UCB 値} = \frac{\text{注目しているノードで勝った数}}{\text{注目しているノードに割いたプレイアウト数}} + c\sqrt{\frac{2\log(\text{全プレイアウト数})}{\text{注目しているノードに割いたプレイアウト数}}}$$

レバー*i*のこれまでの賞金



$$\text{UCB 値} = \frac{Q_n(i)}{n_i} + \sqrt{\frac{2\log n}{n_i}}$$

レバー*i*を引いた数

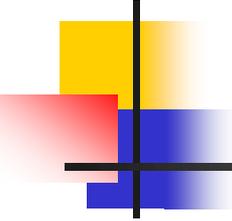


# GA Theory (2)

---

- Deceptive Problem
  - Minimal deceptive problem (MDP)
    - $f_{11}$  が最適
    - $f(0^*) > f(1^*)$  または  $f(*0) > f(*1)$  が成立する
- No Free Lunch (NFL) Theorem

$$\frac{1}{|Y|^{|X|}} \cdot \sum_f P(d_m^y | f, m, a) = \frac{1}{|Y|^{|X|}} \cdot \sum_f P(d_m^y | f, m, a')$$



# GA Search Property

---

- Crossover
  - Global search operator
  - Useful substructures combination
- Mutation
  - Local search operator
  - Schema recovery

# アオカケスによるガ

- 各実験日において、アオカケスにより発見(攻撃)されたガは取り除かれた。
- そして次の日の朝に生き残ったガの集団は相対的な割合を維持されたままもとの数まで戻された
- 生き残り率に比例した世代交代

- 30日間=30世代の実験を繰り返す。
- その結果、隠蔽的なタイプのガが全体の75%を占めて安定した。
- 次に、世代交代の際に突然変異や交叉を導入して明るさや模様の変異を進化させてみた。



鳥が攻撃すると適合度がアップ

鳥が攻撃しないガは次世代に残らない



アオカケス:

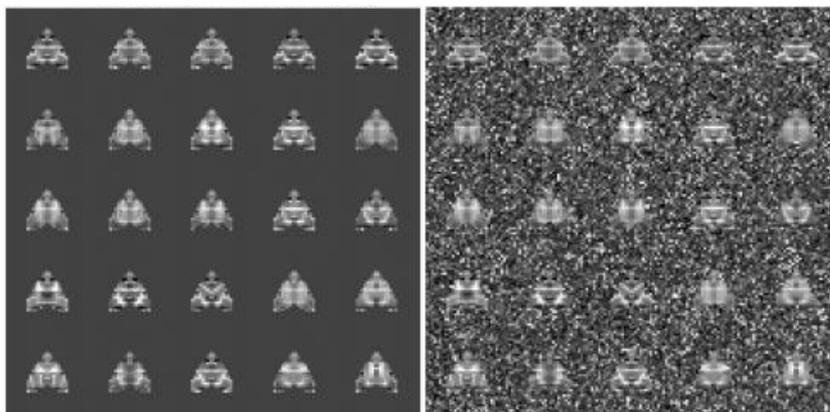
スズメ目カラス科の鳥類。鮮やかな青い羽毛の尾羽を持つ。昆虫や植物の種子や果実などを主な餌とする。

Bond, A.B. and Kamill, A.C.,  
"Visual predators select for crypticity and polymorphism in virtual prey,"  
*Nature*, vol.415, pp.609--613, 2001.

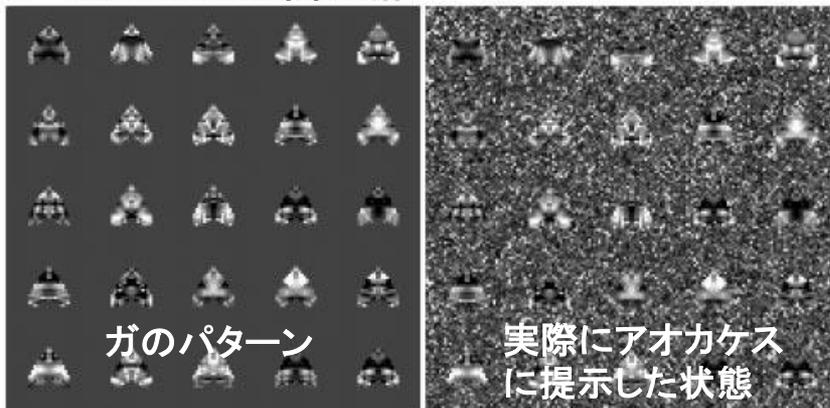
# ガの模様の進化

- アオカケスは突然変異による変則的な隠蔽タイプのガを発見することにしばしば失敗した.
- そのため、このような変異型は頻度が増加した.
- 世代を経るに従って、ガは発見されにくくなり、さらに表現型(模様)が大きくなばらつきを示すようになった.

(a) 初期世代



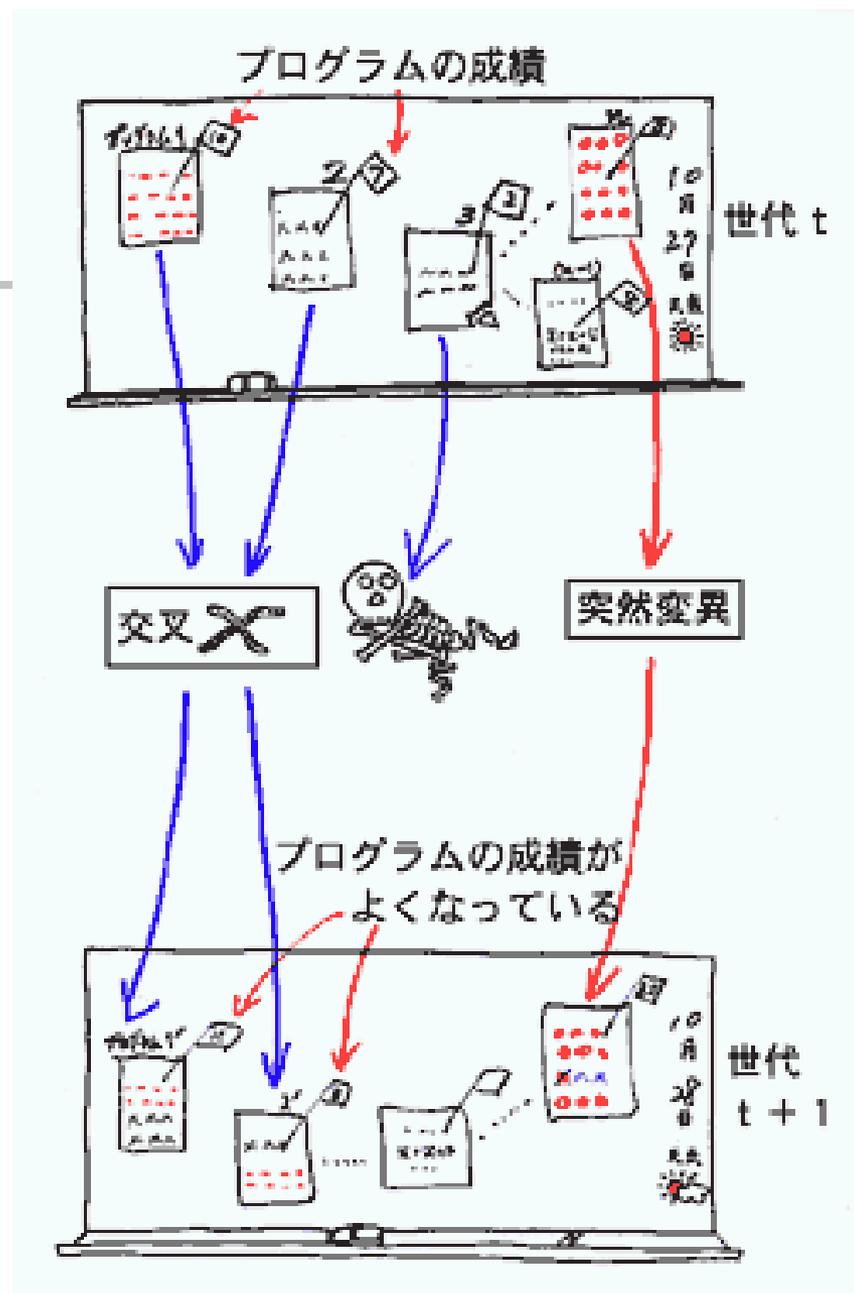
(b) 第100世代



- 自然界で起こる**共進化(軍拡競争)**に類似
- 動物は捕食者から見つかりにくいように隠蔽や擬態のパターンを改善し、多型を増やしていくことが観察されている.

# What is GP?

- GA: パラメータの最適化 (**optimization**)
- GP: プログラムを進化させる
  - 人工知能
  - 問題解決
  - デザイン
  - プログラム合成

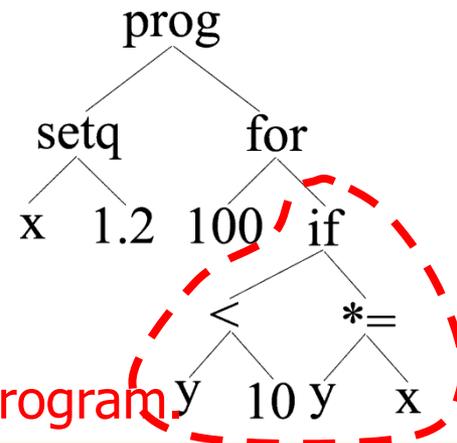


# Genetic Programming (GP)

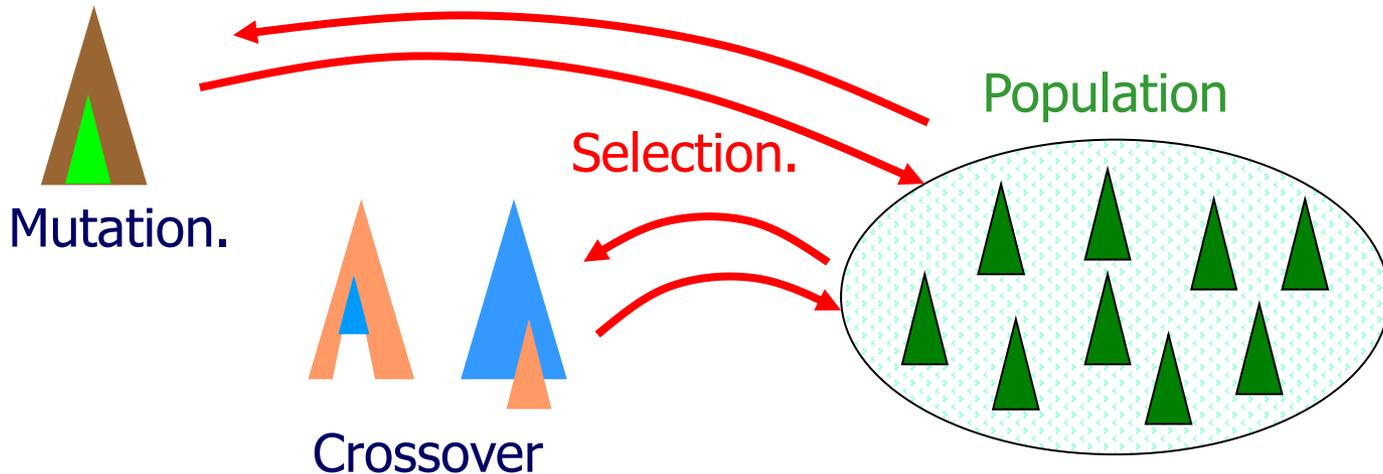
Representation :

```
x = 1.2;  
for (int i=0; i<100; i++) {  
  if (y < 10) {  
    y *= x;  
  }  
}
```

Tree

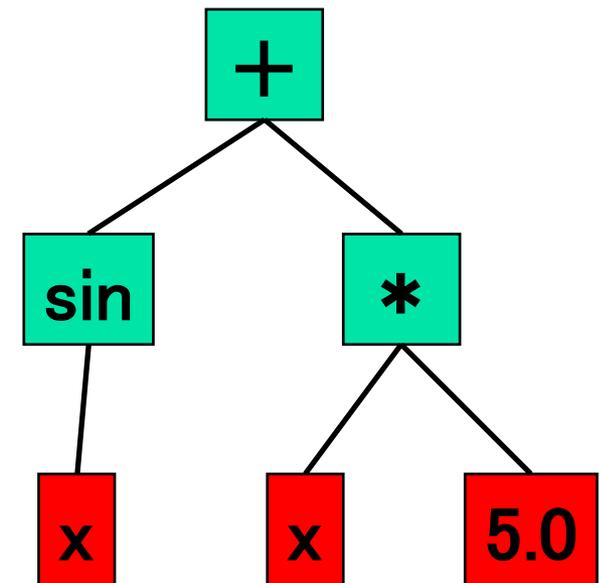


Part tree =  
Substructure of a program.

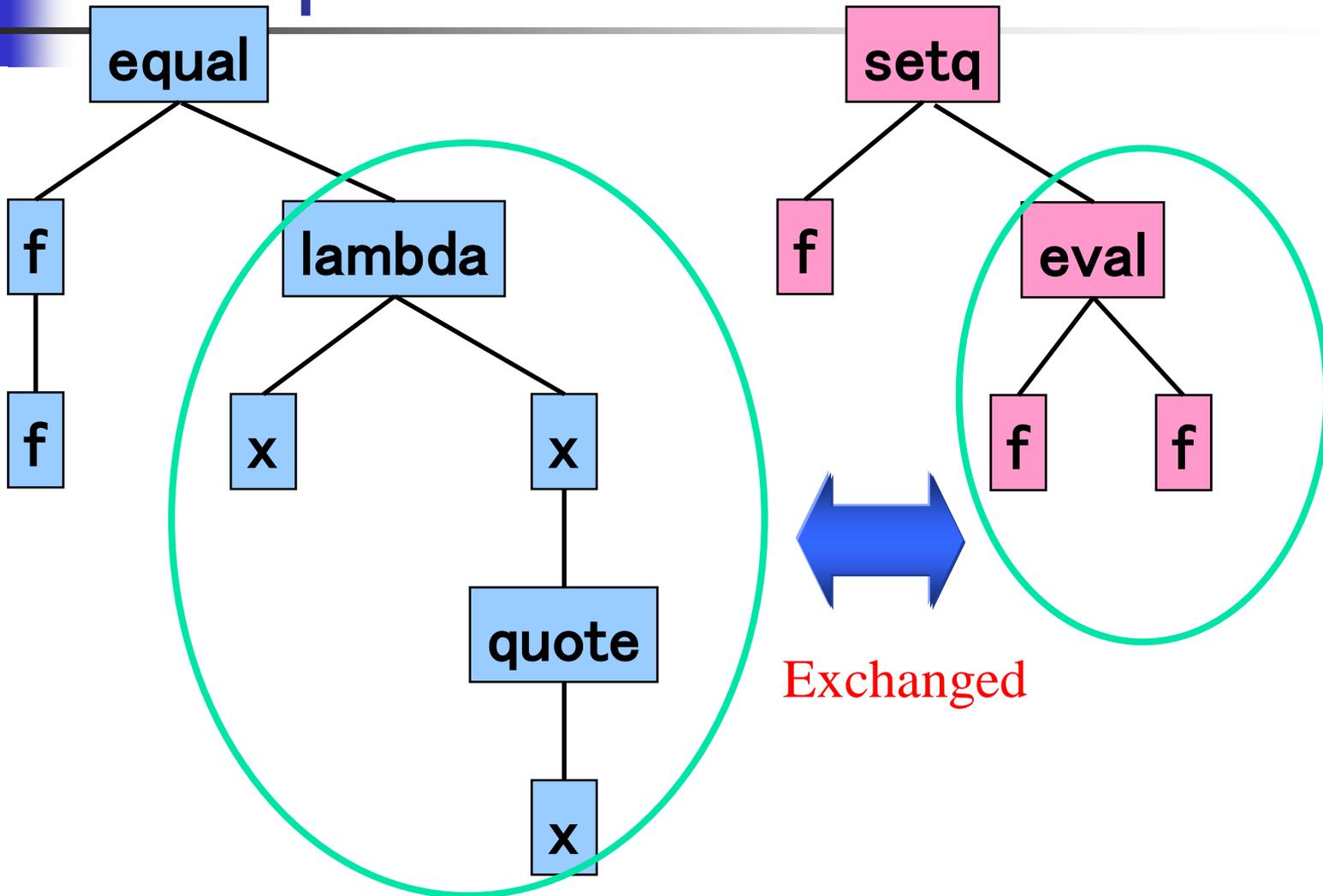


# GPの基本要素

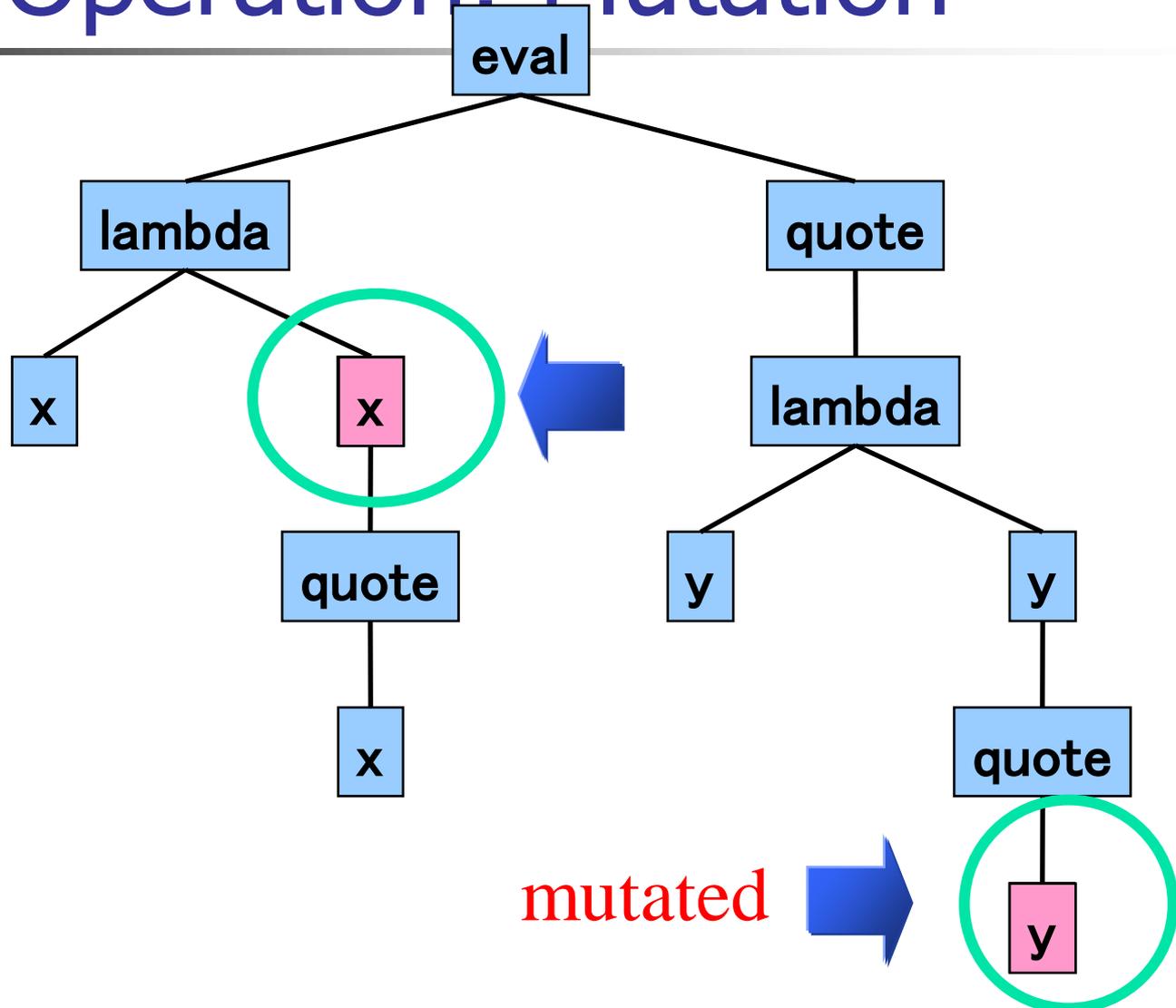
- **Terminals** (終端記号、定数、引数)
- **Functions** (非終端記号、関数)
- Fitness (適合度関数)
- Parameters (パラメータ)
- Termination criterion



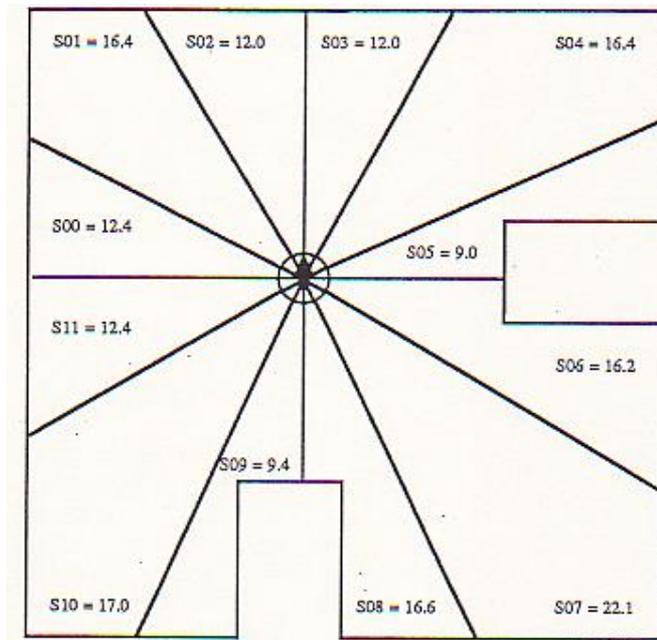
# GP Operation: Crossover



# GP Operation: Mutation



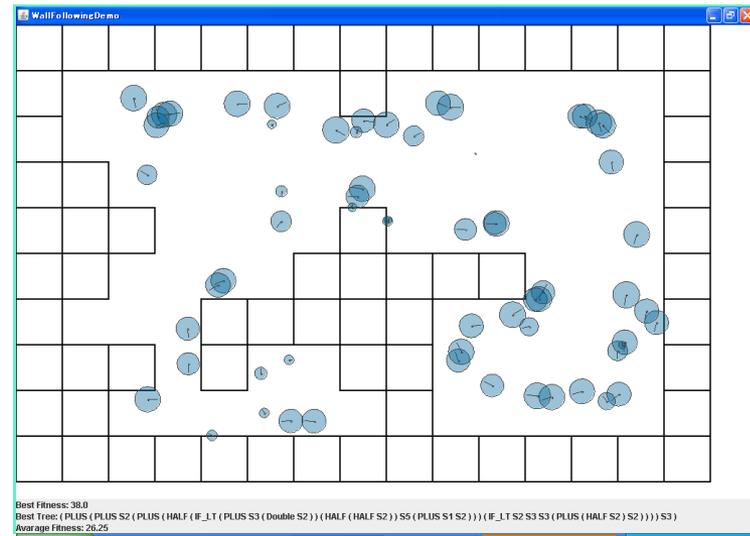
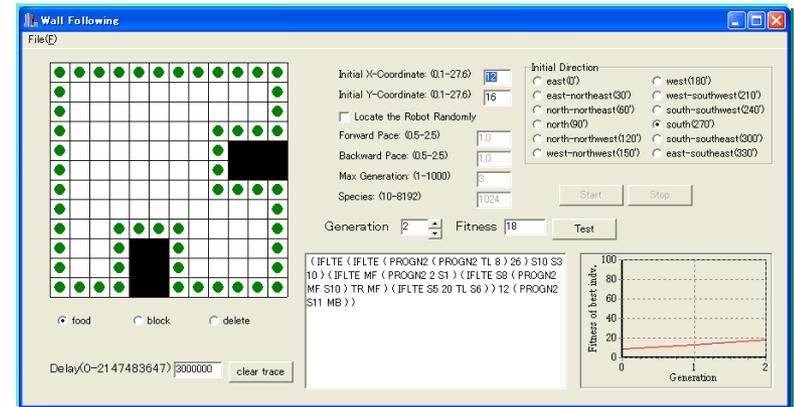
# Example: Wall Following Task



- $F = \{TR, TL, MF, MB, IFLTE, PROG2\}$
- $T = \{S00, S01, \dots, S11, SS, MSD, EDG\}$

# デモ

- Swam version
  - Wall following
  - Box moving



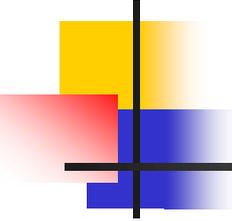
# いまなぜGA/GPか？

- 最適化ではない進化手法への期待
  - プログラム設計、デザイン
  - そこそこで早い

- 高速処理技術の確立
  - 並列化
  - 精度向上

- AI revisited
  - コンテンツの扱い
  - 構造的表象、オントロジー、認知発生学、意味論
  - ...





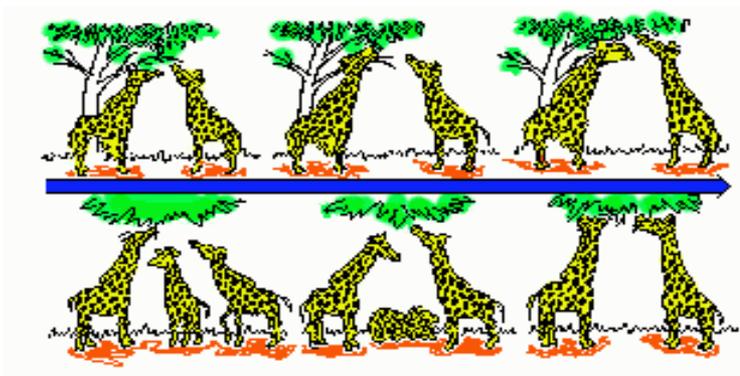
# 何がGA/GPで問題か？

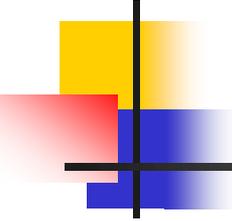
---

- 進化とは何か？
- もっとよい学習方法はないか？
- 理論的な背景はないか？

# GAから交叉と突然変異を無くしたら？

- GAによる最適化
  - 集団による多様性維持
  - 選択
  - 交叉と突然変異による遺伝子の置き換え

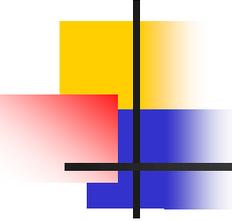




# EDAとは何か？

---

- **EDA=Estimation of Distribution Algorithm**
- 進化計算 (Evolutionary Computation) における新しい考え方
- **機械学習、統計、クラスタリング**などの技法と**集団学習の統合**
  - **EDA + VAE**
  - **EDA + DL**

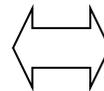


# EDAとは何か？

---

## Evolutionary Computation

Reproduction  
Operators (Crossover  
& Mutation)



## EDA

Probabilistic Model  
Building and Sampling

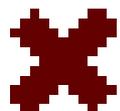
- **Selection** と **Replacement Strategy** は進化計算と同じである

# GA v.s. EDA

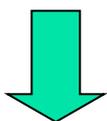
## Genetic Algorithm

Parents

1 1 0 1 0 1 0 1 0 1



0 1 0 1 0 1 1 1 0 0



Offspring

1 1 0 1 0 1 1 1 0 0

0 1 0 1 0 1 0 1 0 1

## EDA

Parents

1 1 0 1 0 1 0 1 0 1

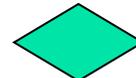
0 1 0 1 0 1 1 1 0 0

確率分布



0.5 1 0 1 0 1 0.5 1 0 0.5

乱数発生 (一様分布など):



0.6 0.8 0.1 0.4 0.9 0.5 0.4 0.1 0.2 0.9

Offspring



0 1 0 1 0 1 1 1 0 0

# EDAの簡単な例

3 遺伝子 :  $x_0 x_1 x_2$

$$p(x_0 = 1) = 0.75$$

$$p(x_1 = 1) = 0.75$$

$$p(x_2 = 1) = 0.5$$

Generate

110 011

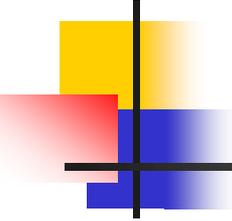
110 111

010 111

110 110

Estimate  
Distribution

Sampling



# Probabilistic Model Building EA

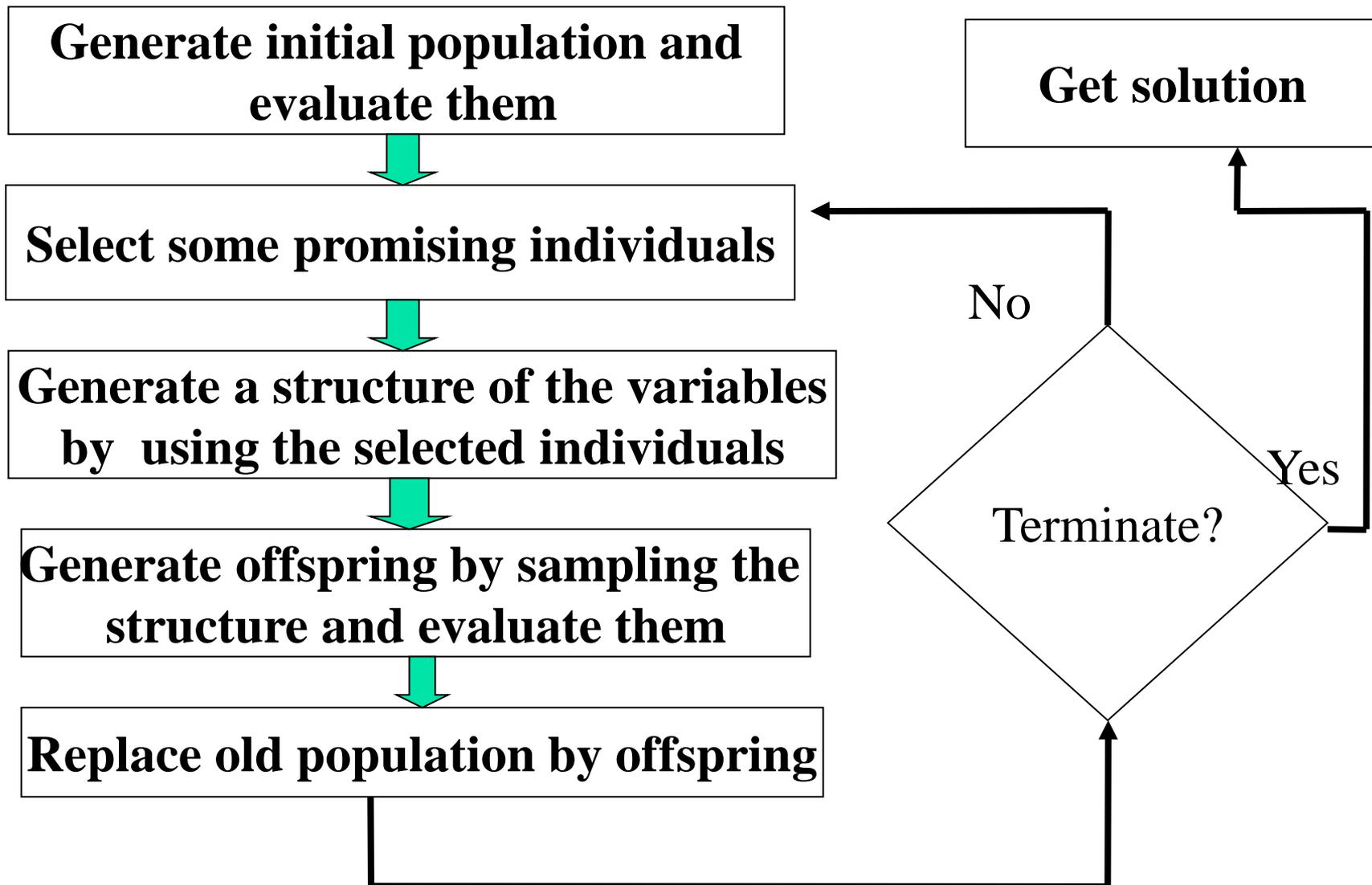
---

- $p(\mathbf{X} \mid \mathbf{D})$ を求めることが目的
  - さまざまな確率モデルが考えられる
    - 独立, 二変数依存, 同時確率, 多変数依存
- 求めた $p(\mathbf{X} \mid \mathbf{D})$ によって新たに個体を生成

XはChromosome

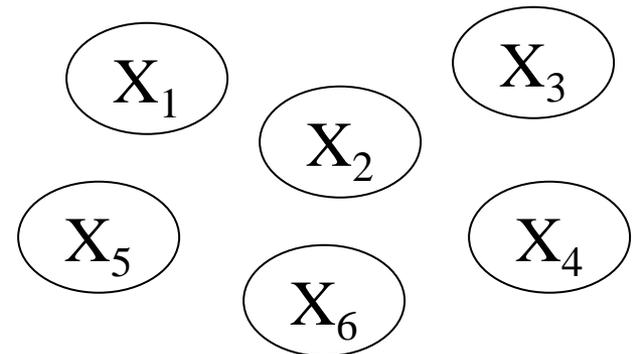
Dは選択された優良個体

# EDAのアルゴリズム



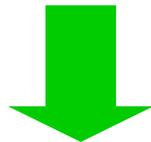
# Univariate EDAs in Discrete Domain

- **UMDA** (Muehlenbein, 1998)
  - **Population-based**
- **PBIL** (Baluja, 1994)
- **cGA** (Harik et al., 1998)
  - **probability vector**
  - 異なる更新規則



# 独立: UMDA

- 各変数を独立とする
- パラメタは最尤推定により求められる
- 変数間の依存関係は推定されない



より複雑な変数間のモデル  
構造を推定する

$$p(\mathbf{X} | D) = p(\mathbf{X} | \boldsymbol{\theta})$$

$$= \prod_{i=0}^{n-1} p(x_i | \boldsymbol{\theta})$$

$$= \prod_{i,j} \theta_{ij}$$

$\theta_{ij}$  :  $i$ 番目の遺伝子が

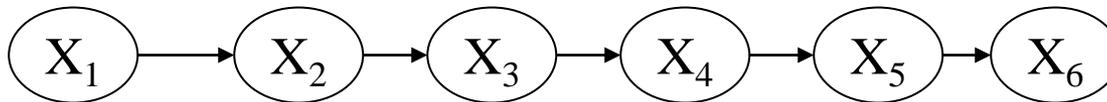
$j$ 番目の状態にある確率

$$\boldsymbol{\theta} = \arg \max_{\boldsymbol{\theta}} \{l(\boldsymbol{\theta} | D)\}$$

$$= \arg \max_{\boldsymbol{\theta}} \{p(D | \boldsymbol{\theta})\}$$

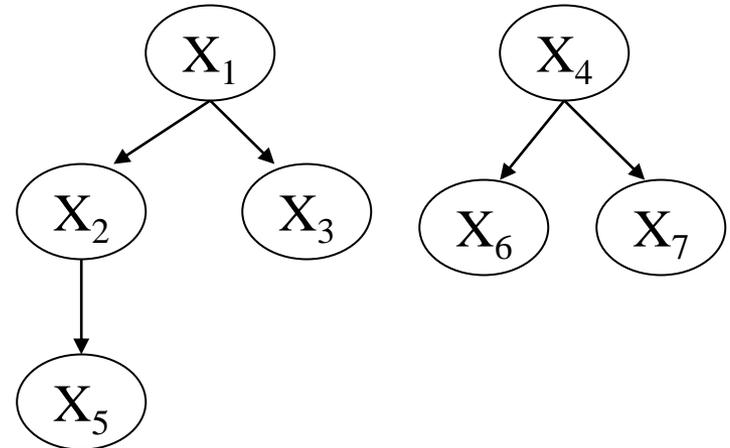
# Bivariate EDAs in Discrete Domain

**MIMIC** (De Bonet et al., 1997):



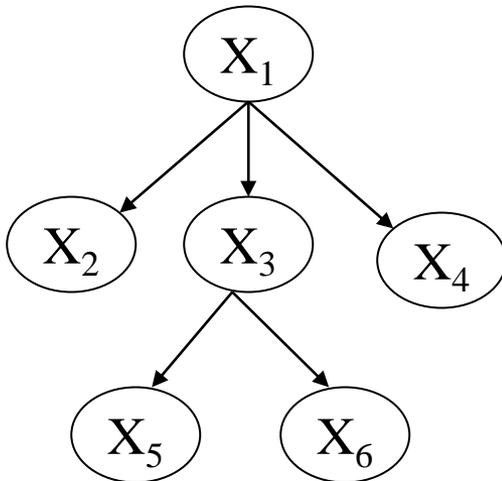
(a chain of nodes)

**BMDA** (Pelikan and Muehlenbein, 1999):



(a forest of trees)

**COMIT** (Baluja and Davies, 1997):

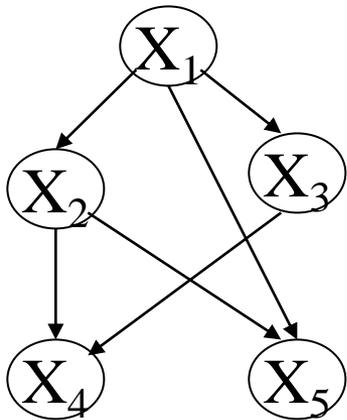


(Tree Structure)

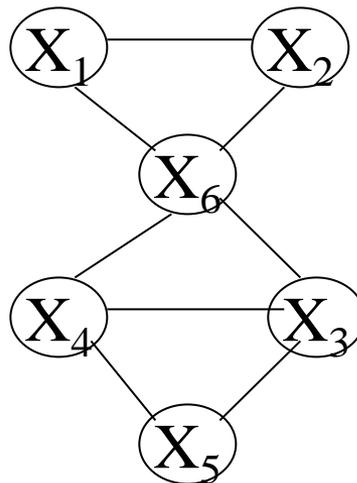
# Multivariate EDAs in Discrete Domain

**EBNA** (Larrañaga et al., 2000)

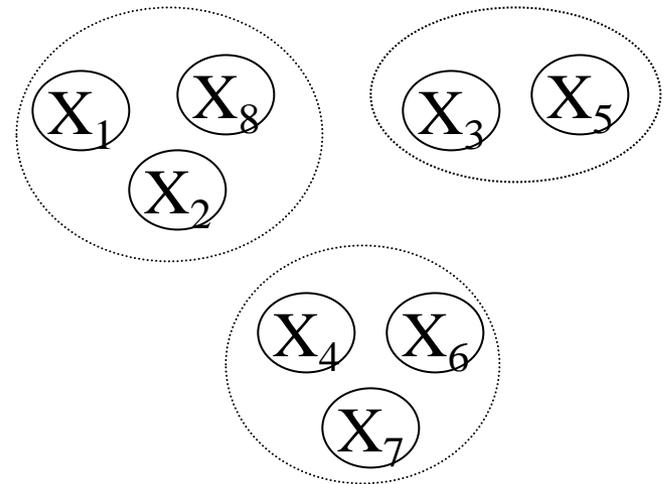
**BOA** (Pelikan et al., 2000)



**FDA** (Muehlenbein and Mahnig, 1999)



**EcGA** (Harik, 1999)



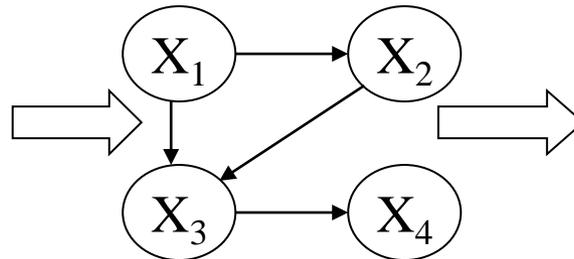
**RELEDA** (Topton & Iba 03,04)

**RELEDA = 強化学習 + EDA**

# 遺伝子連鎖の推定

$X_1$	$X_2$	$X_3$	$X_4$	F
1	0	1	1	3
1	1	0	1	3
0	1	1	0	2
0	1	0	0	1

Structure Learning



$$P(X_1) = 0.5$$
$$P(X_2|X_1) = 0.5$$
$$P(X_3|X_1, X_2) = 0$$
$$P(X_4|X_3) = 0.5$$

Offspring

$X_1$	$X_2$	$X_3$	$X_4$	F
1	1	1	1	4
1	1	0	1	3
0	1	1	0	2
1	0	0	1	2

**Trade-off:**

- 推定する分布が一般的になるほど、計算量(必要なサンプル数、推定時間)が増える

# 二変数依存:MIMIC

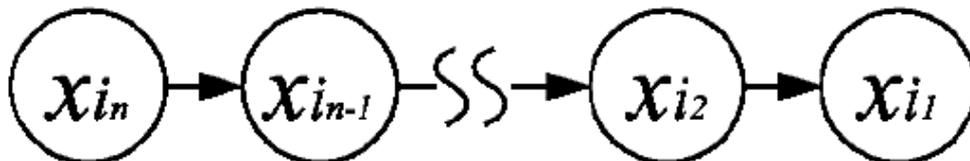
- 二変数依存考慮
- 直鎖構造
- 真の分布を最もよく近似する直鎖の順列nを探索
  - KL距離を最小化する順列

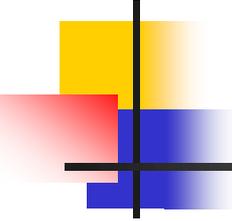
$$\pi = \arg \min_{\pi} \left\{ KL(p(\mathbf{X} | D) \parallel p(\mathbf{X} | \pi)) \right\}$$

$$p(\mathbf{X} | D) \approx p(\mathbf{X} | \pi)$$

$$= \prod_{i=1}^{n-1} p(x_{\pi_i} | x_{\pi_{i-1}})$$

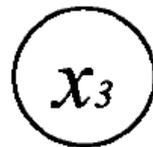
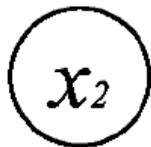
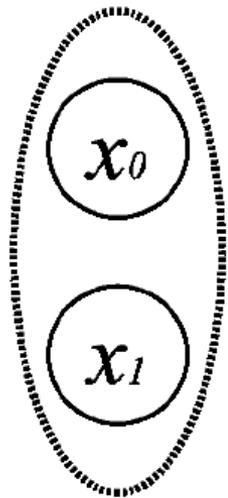
$$= \prod_{i,j,k} \theta_{ijk}$$





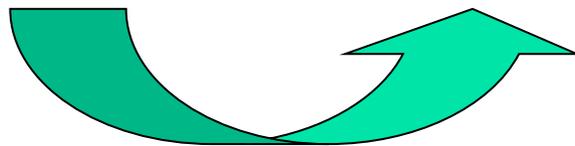
# 同時確率: ECGA

- 最小記述長 (MDL) によって, 同時確率のパーティションを決定する
- 同時確率で変数の依存関係を推定



$$p(\mathbf{X} | D) = p(X_0, X_1, X_2, X_3 | \boldsymbol{\theta})$$

$$= p(X_3 | \boldsymbol{\theta}_3) p(X_2 | \boldsymbol{\theta}_2) p(X_1, X_0 | \boldsymbol{\theta}_{0,1})$$



# BOA, EBNA, LFDA

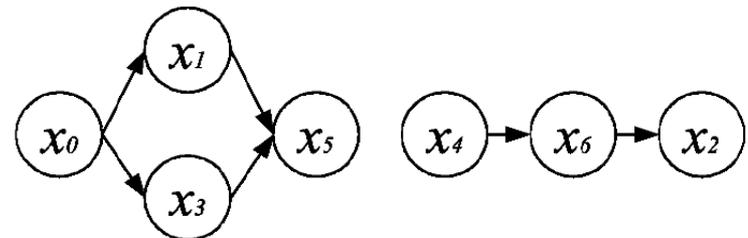
- 多変数依存
- $p(M | D)$ を最大にする $M$ を探索
  - BIC, BD metric + K2 algorithm, Algorithm B

$$p(\mathbf{X} | D) = \sum_M p(\mathbf{X} | M, D) p(M | D) \quad M = \arg \max_M p(M | D)$$

$$\approx p(\mathbf{X} | M, D) p(M | D)$$

$$\approx p(X | M, D)$$

$$= \prod_i p(x_i | \pi_i)$$



# EDP (Estimation of Distribution of Programming)

- 集団内の優秀な個体の分布
- 推定された分布に基づいて新しい個体を生成

- 交叉、突然変異を賢くしたい
- Bloatを抑えたい
- Intronの成長を制御したい

交差や突然変異がない進化

*Distribution*

*Gene*

*Frequency*

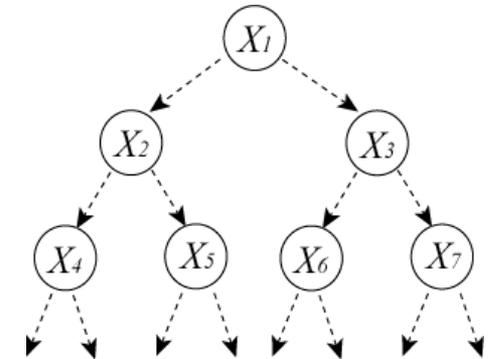
*Generate the new population based on the distribution*

# 確率分布モデル



## ■ 確率木

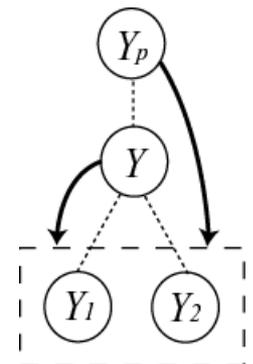
- プログラムの大域的な構造を推定
- 木構造のベイジアンネット
- 1つ上の親のみに従属



## ■ 再帰分布

- 有用な部分構造を推定
- 1つ上の親と2つ上の親のノードのシンボルを条件とする子ノードの条件付同時確率
- 位置に独立した推定

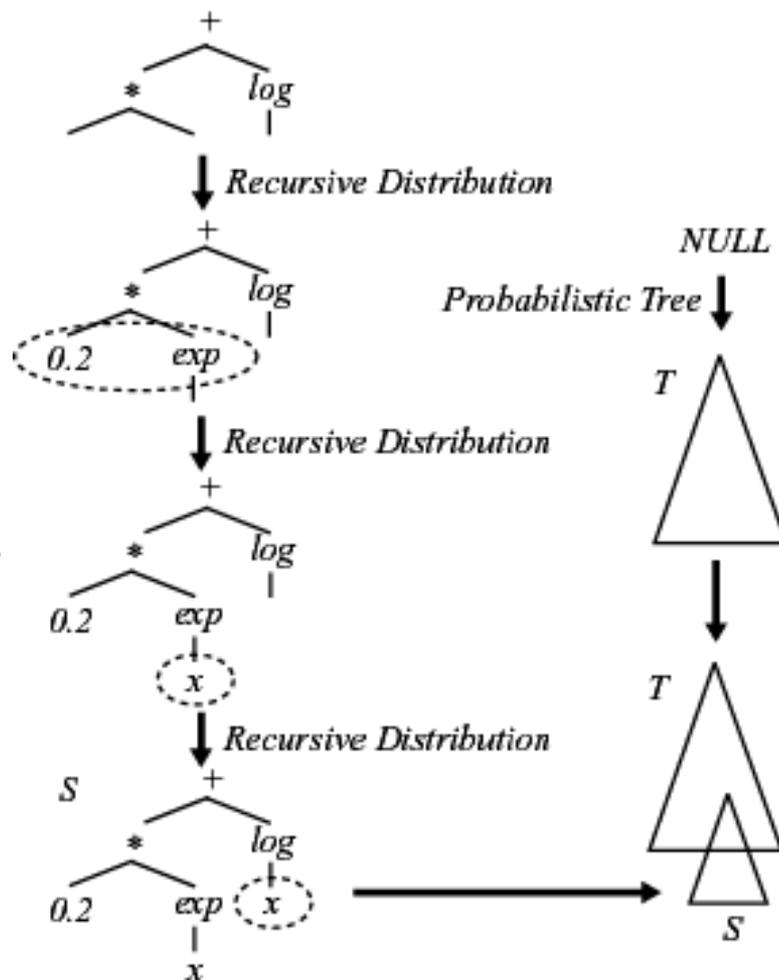
$$P(Y_1, Y_2, \dots, Y_{a_{max}} | Y, Y_p)$$



# プログラムの生成

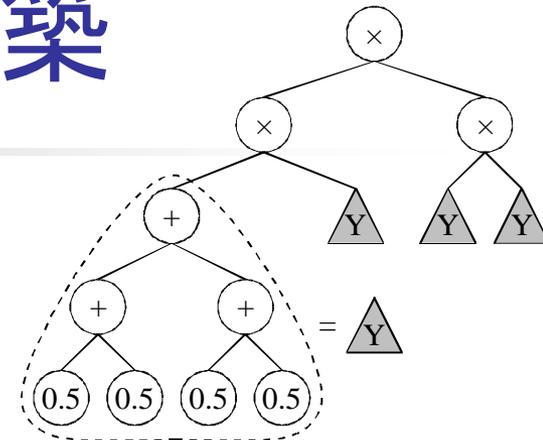
- 1. 確率木に従ってプログラムTを生成
- 2. 再帰分布に従って部分木Sを再帰的に生成
- 3. Tの任意の部分木をSで置き換える

前の個体を引き継がず、  
毎回、0から新しく個体を生成する。

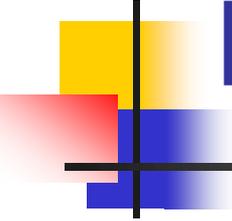


# プログラム進化の再構築

- Probabilistic Function Evolution based on Estimation of Distribution
  - **POLE**
    - Bayesian network + expanded parse tree
  - **PAGE-EM**
    - PCFG with latent annotations + EM Algorithm
  - **PAGE-VB**
    - PCFG with latent annotations + Variational Bayes



確率分布木の構成とその推定方法により  
さまざまなバリエーションがある



# Parameter Estimation with EM

---

- EM Algorithm

- Maximize the lower bound of log-likelihood
- Iterative optimization (hill-climbing)

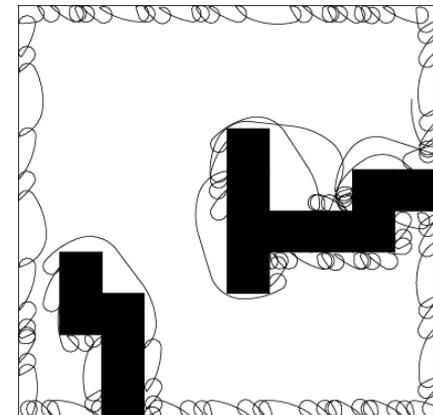
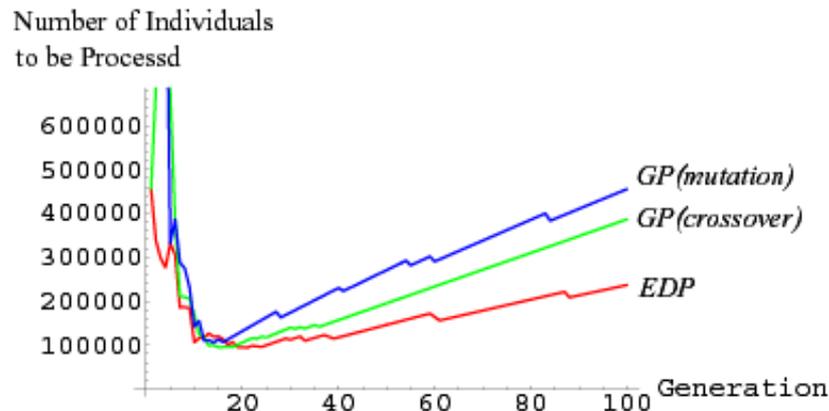
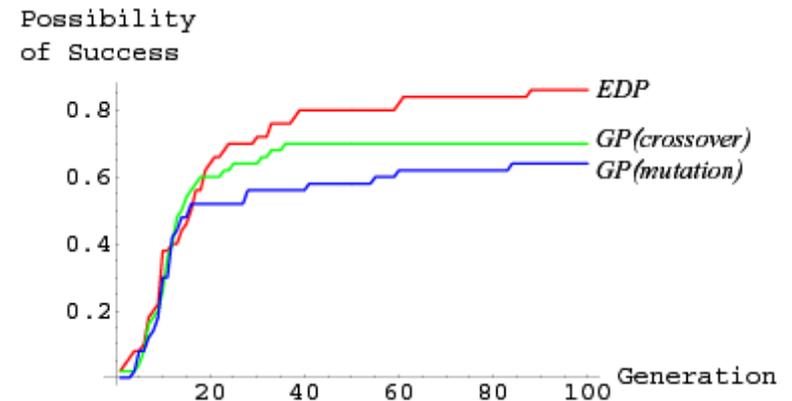
$$\begin{aligned} & \log P(T; \Theta') - \log P(T; \Theta) \\ &= \log \frac{P(T; \Theta')}{P(T; \Theta)} \\ &= \sum_{\mathbf{x}} P(\mathbf{x}|T; \Theta) \log \frac{P(T, \mathbf{x}; \Theta')}{P(T, \mathbf{x}; \Theta)} \frac{P(\mathbf{x}|T; \Theta)}{P(\mathbf{x}|T; \Theta')} \\ &\geq \sum_{\mathbf{x}} P(\mathbf{x}|T; \Theta) \log \frac{P(T, \mathbf{x}; \Theta')}{P(T, \mathbf{x}; \Theta)} \end{aligned}$$

$$Q(\Theta'|\Theta) = \sum_{T_i \in \mathbf{T}} \sum_{\mathbf{X}_i} P(\mathbf{X}_i|T_i; \Theta) \log P(T_i, \mathbf{X}_i; \Theta')$$

# Wall-following Problem (cont.)

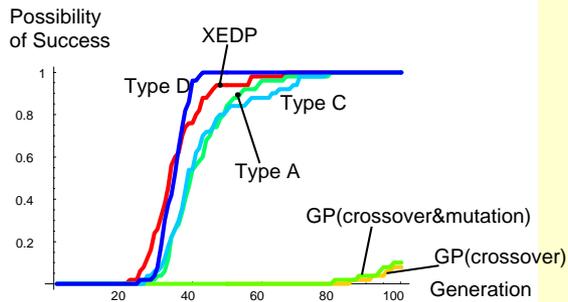
- 進化の後半においてもEDPは進化し続ける。
  - 広い範囲に渡り、EDPはGPよりも小さい  $I(1000, i, 0.99)$  を持つ

○

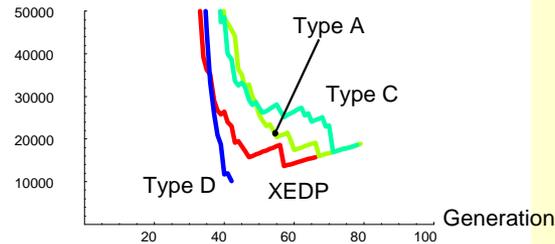


# 比較実験

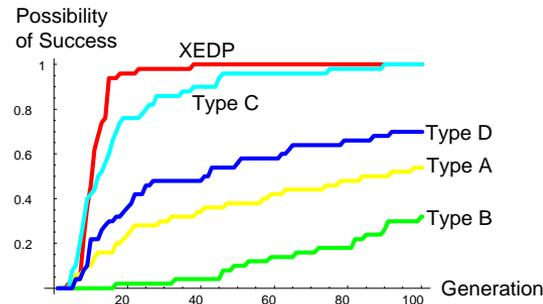
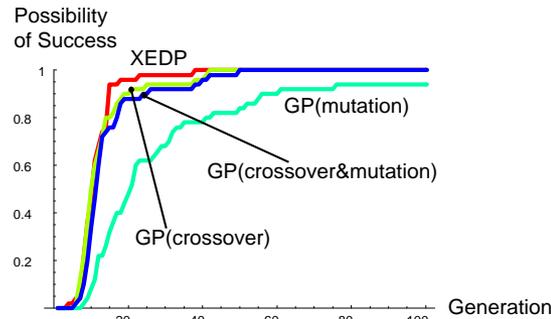
## Max problem



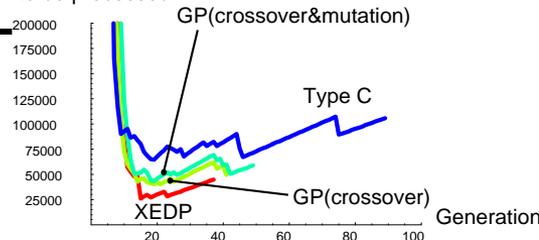
Number of Individuals to be processed



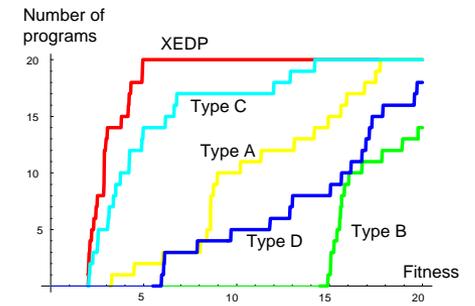
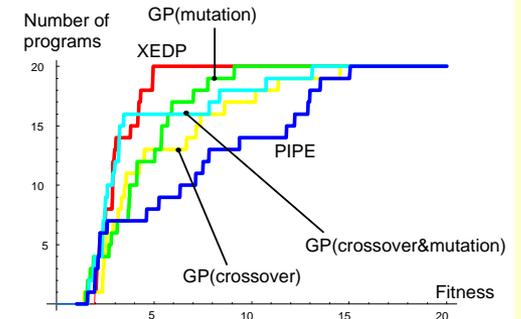
## Multiplexer problem



Number of Individuals to be processed



## Regression problem



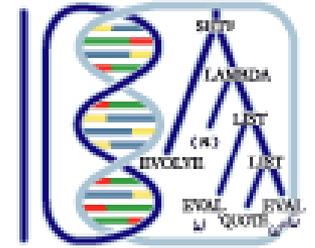
(The best fitness at 50<sup>th</sup> generation)

## Related methods

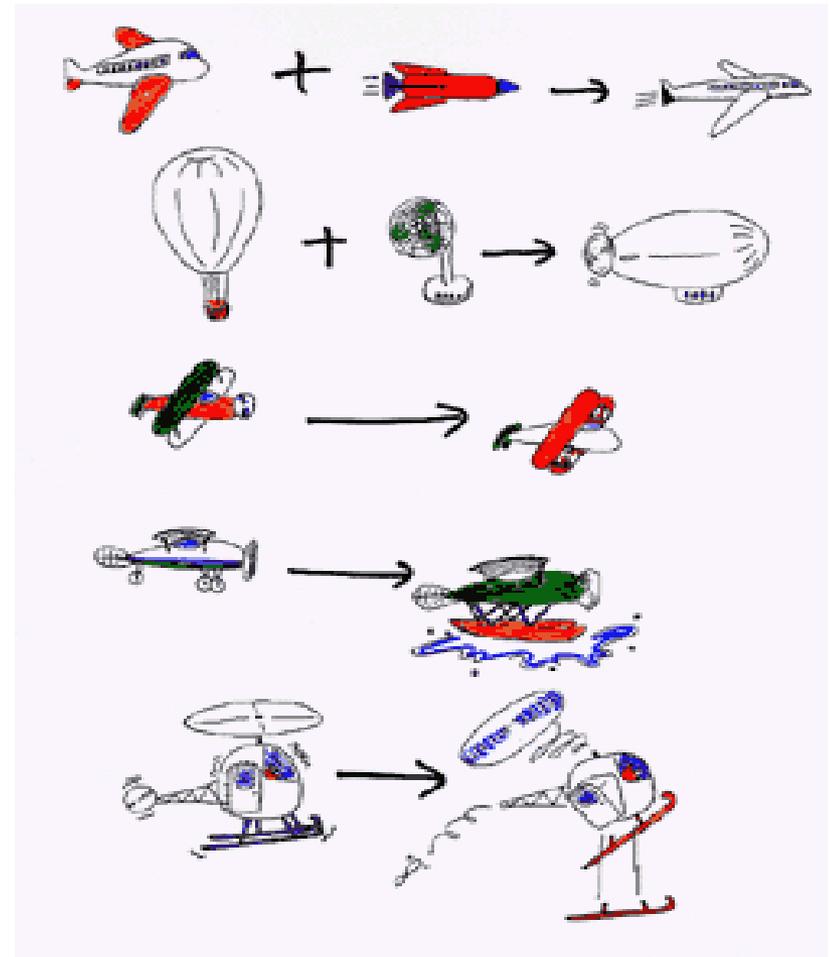
- Type A** Uses only the CPT.
- Type B** Uses only the RD.
- Type C** Uses the CPT and random mutation.
- Type D** Uses the PPT\* and the RD.

\* The PPT is the probabilistic model without any probabilistic dependency.

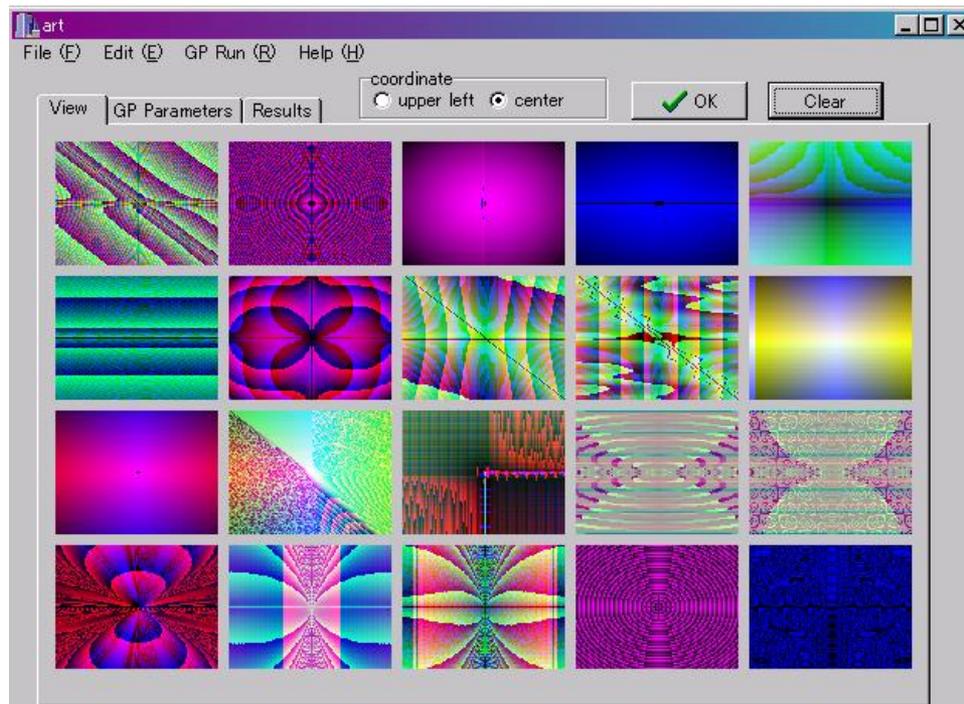
# GAとGPの研究テーマ



- 進化するハードウェア
- 進化型ロボット
- マルチエージェント学習
- 高速化GPシステム
- 金融工学と売買システム
- No Free Lunch Theorem
- 芸術への応用
- 並列化実装
- 量子計算
- . . . . .



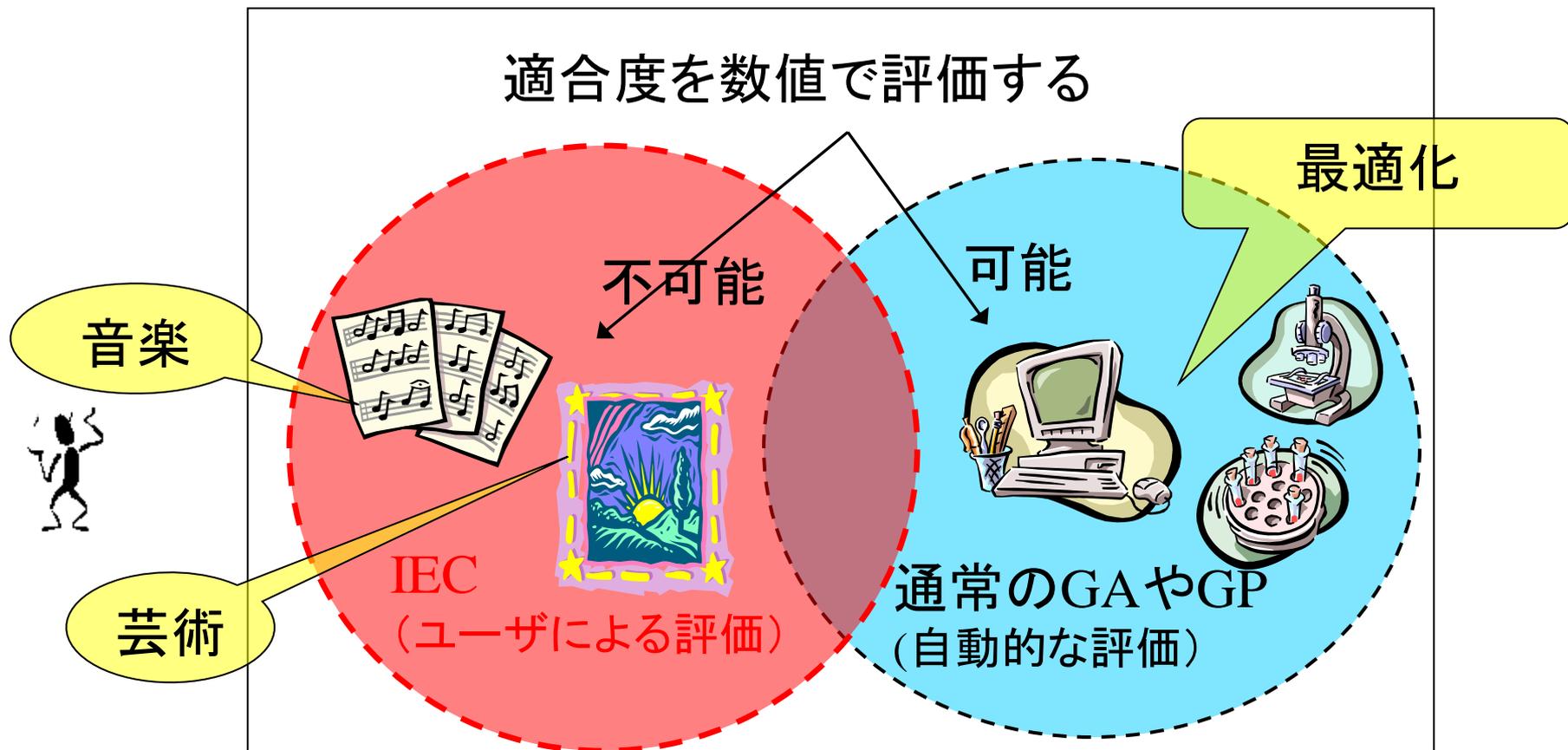
# 抽象画を進化させる?



Answer : 対話型進化計算

IEC: Interactive Evolutionary Computation

# Interactive Evolutionary Computation



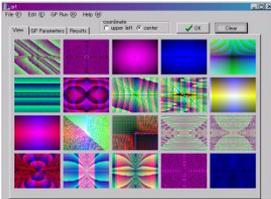
IEC

START

Initialize population

System shows the individuals

Evaluate fitness of populations **by user**

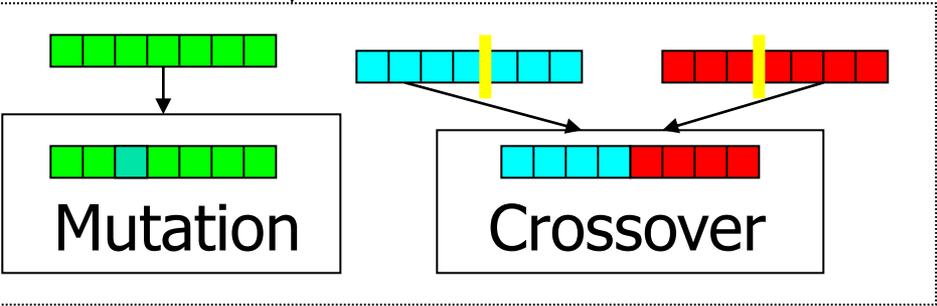


Check terminal

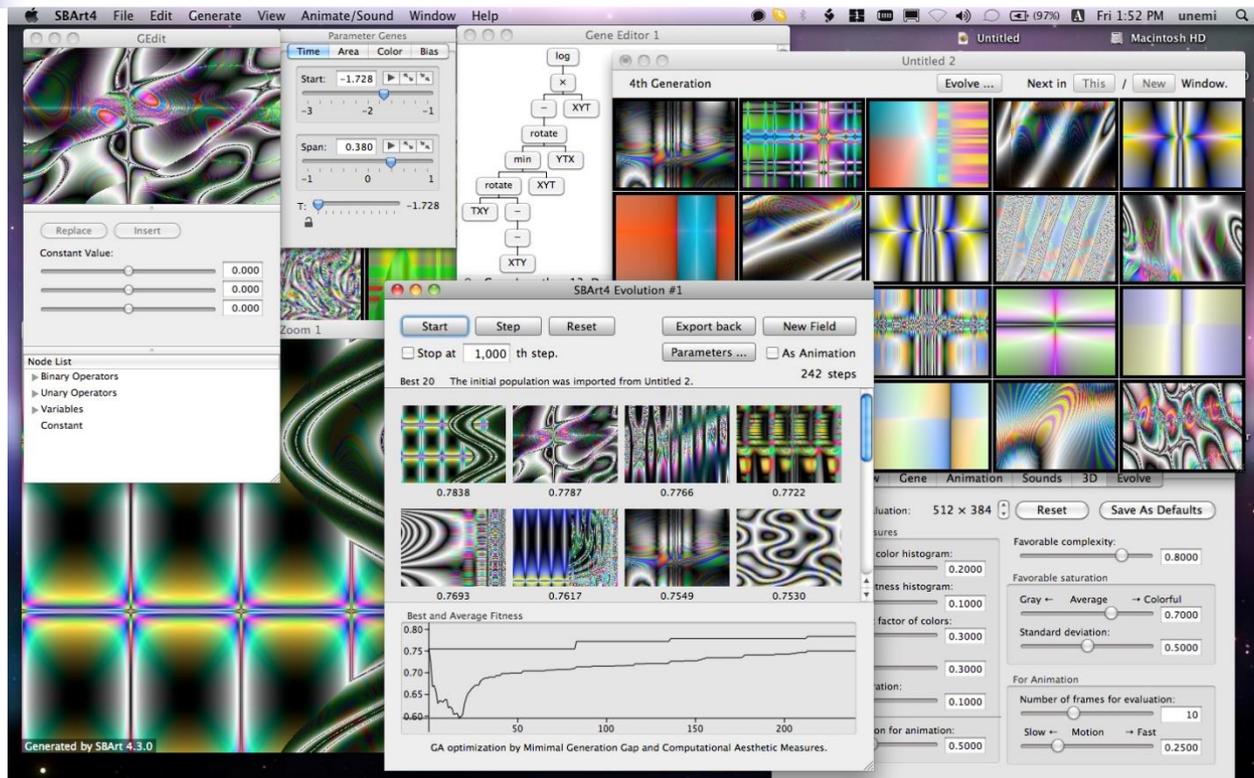
Goal

Selection

Breed new generations



# SBART: 抽象画、ビデオ画像

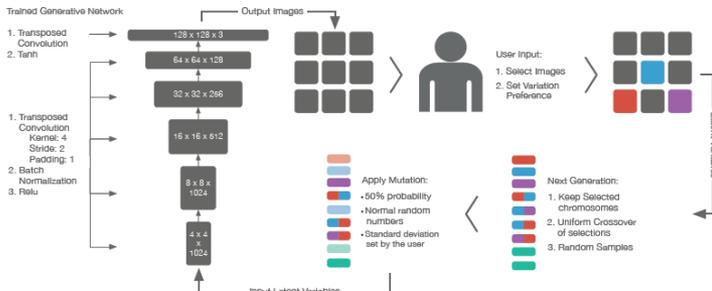
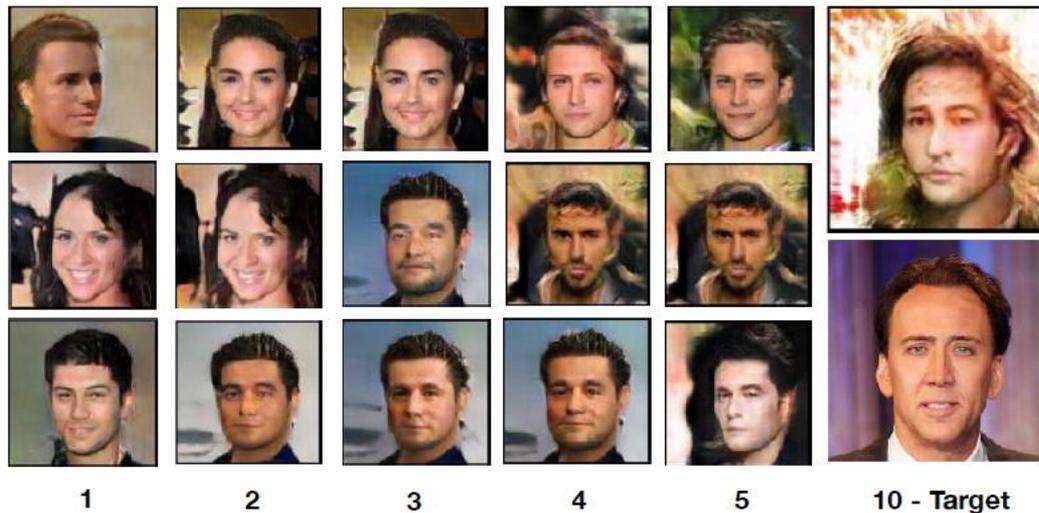


**SBART:** Tatuso Unemi @Soka University

<http://www.intlab.soka.ac.jp/~unemi/sbart/4/ja/>

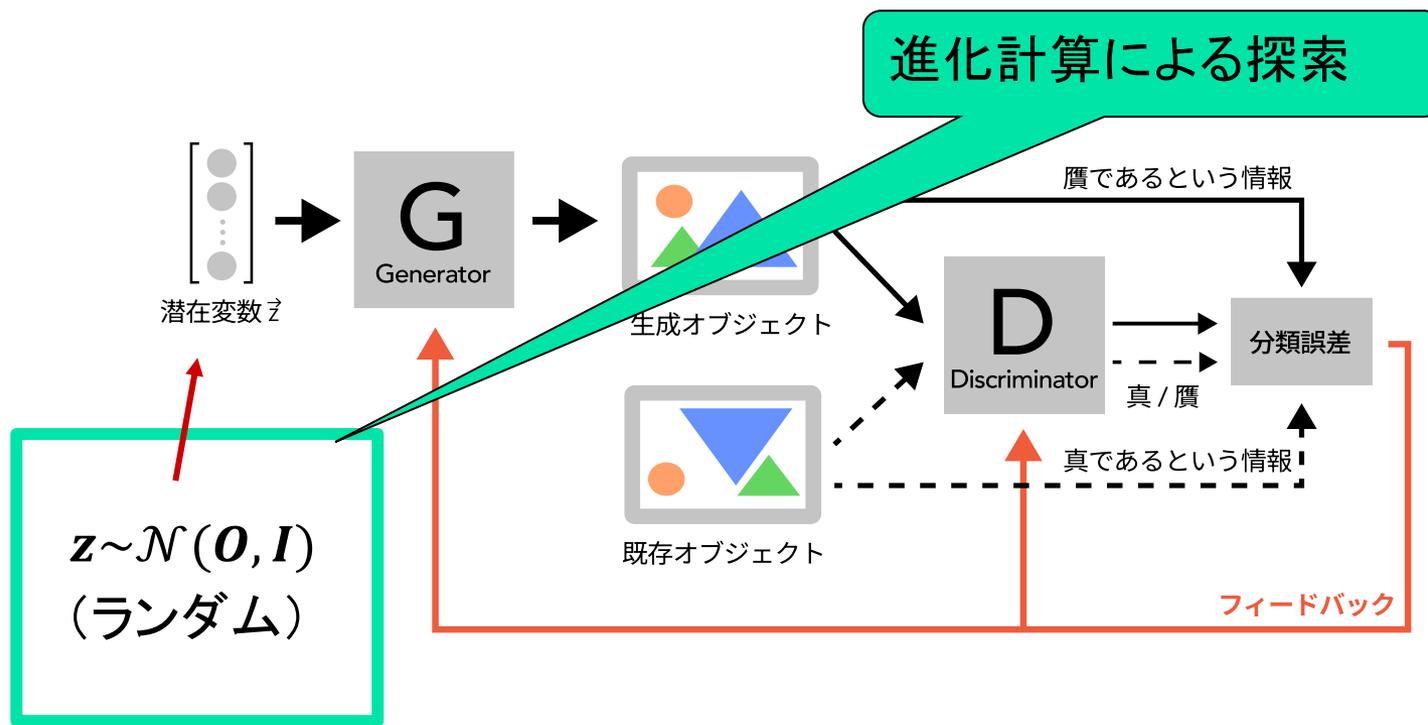
# Deep Interactive Evolution

## ■ Deep learning + IEC



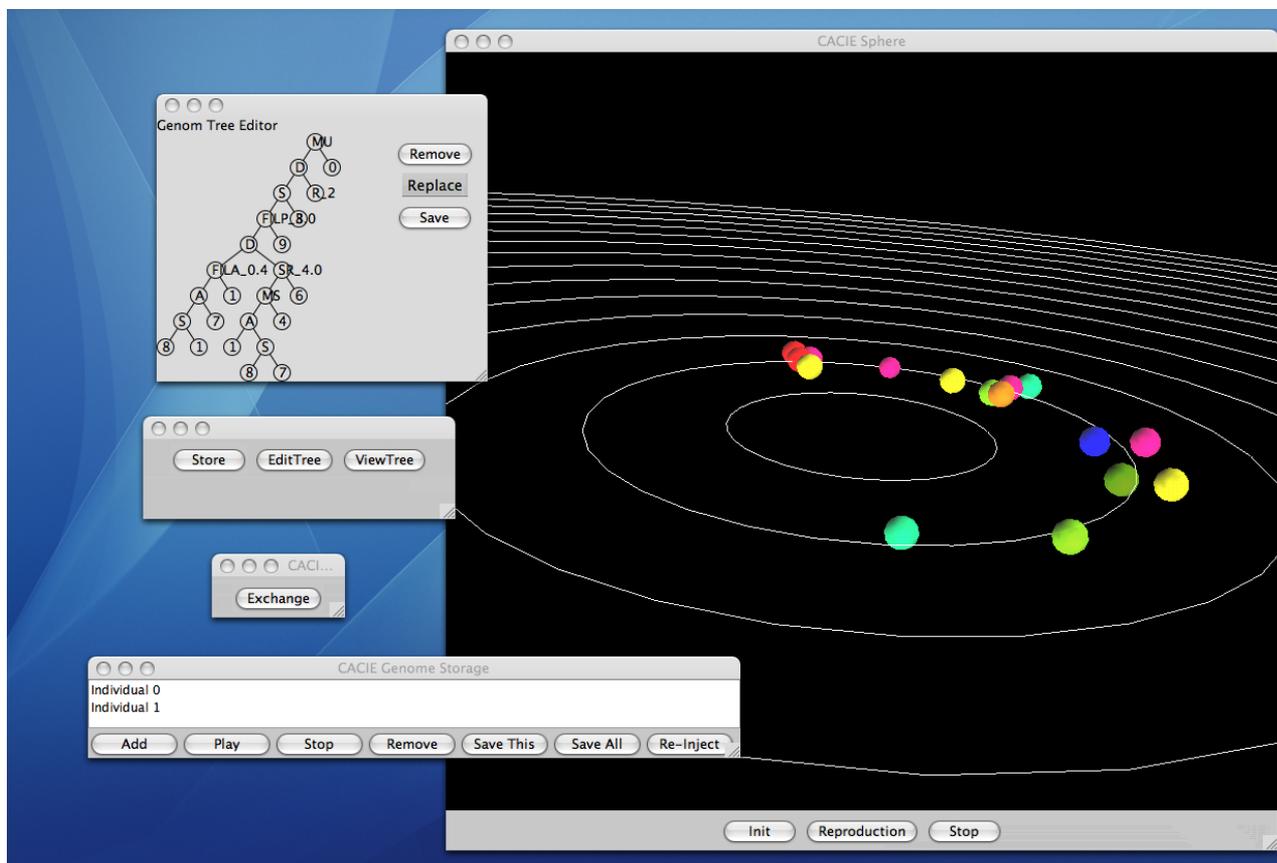
*Deep Interactive Evolution*  
 Philip Bontrager, Wending Lin,  
 Julian Togelius, Sebastian Risi  
 arXiv:1801.08230, 2018.

# GAN for DeepIE



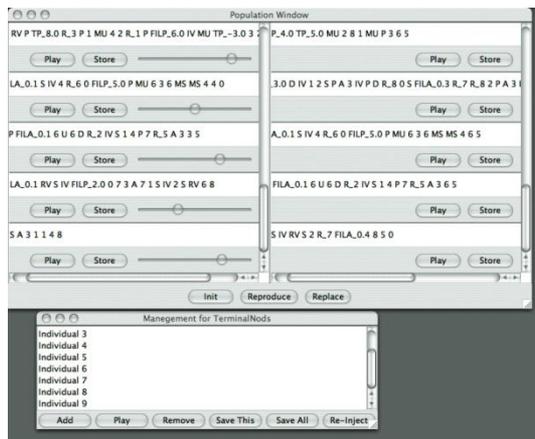


# CACIE:IECによる作曲支援



# 対話型進化手法による作曲

- IEC (Interactive Evolutionary Computation)
- モダン・クラシックのピアニストとの競演



# 現代アートとは何か？

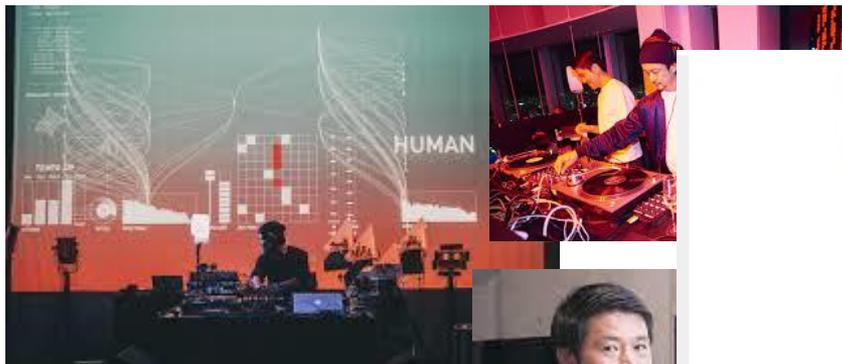
- 今日の芸術家はもはや生産しない。
- あるいは少なくとも生産することが一番重要なのではなく、芸術家は**選別し、比較し、断片化し、結合し、特定のものをコンテクストのなかへ入れ、他のものを除外する**のである。





# 進化計算、AIと芸術

AIはフロアを盛り上げることはできるか？  
みずから生み出した人工知能とコンビでDJをする、Qosmo徳井直生さんに訊く



羽ばたく起  
東京大学工  
徳井直生  
取締役 | E

WIRED JAPANの連載「AWAY FROM ANIMALS AND MACHINES - 動物と機械から離れて」にて徳井直生のインタビューが公開されました。イスラエル・ガルバン+YCAMダンス公演「Israel & イスラエル」を中心に、人間の創造性を拡張するようなAIと人間の関係性など様々な考えについて語っています。



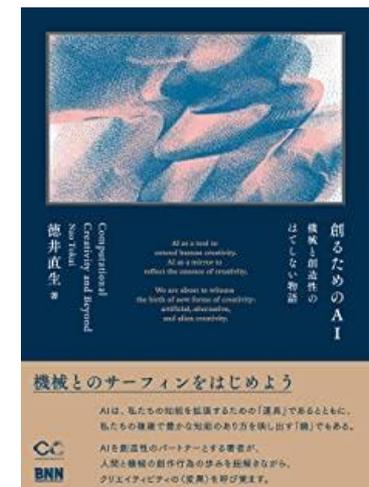
# AIと創造性

- The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities
  - <https://arxiv.org/abs/1803.03453>

- 「創るためのAI 機械と創造性のはてしない物語」、徳井直生、ビー・エヌ・エヌ、2021

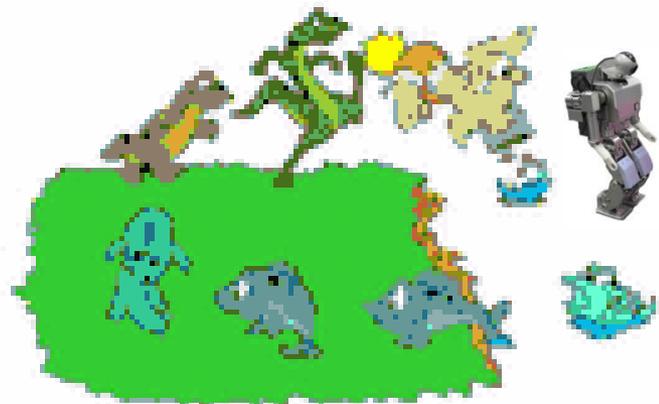
[創るためのAI 機械と創造性のはてしない物語 | 徳井直生 | 本 | 通販 | Amazon](#)

創造性とは何か？



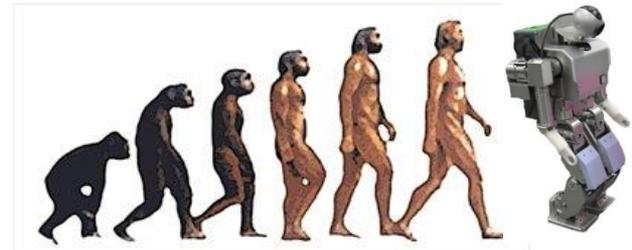
# まとめ

- 進化型ロボットへ
  - NN、Q. . .  
基本的に山登り探索、  
low-level behavior
- GA, GP, AL. . .
  - 確率的大域探索(はったり探索)
  - 探索空間がill-defined
  - 部分構造から全体をさがす
  - 頑強さ、集団探索
- なぜ進化にこだわるのか



# 進化論的手法の特長

- 集団性
  - 工学的意義: 高度に並列化可能
- 多様性
  - さまざまな個性を集団内に維持する
- 共生
  - 相互に影響を与えながら複数の種が進化する
- 進化 ≠ 進歩
  - S.J.Gould: 断続平衡仮説
  - 「停滞はデータなり」

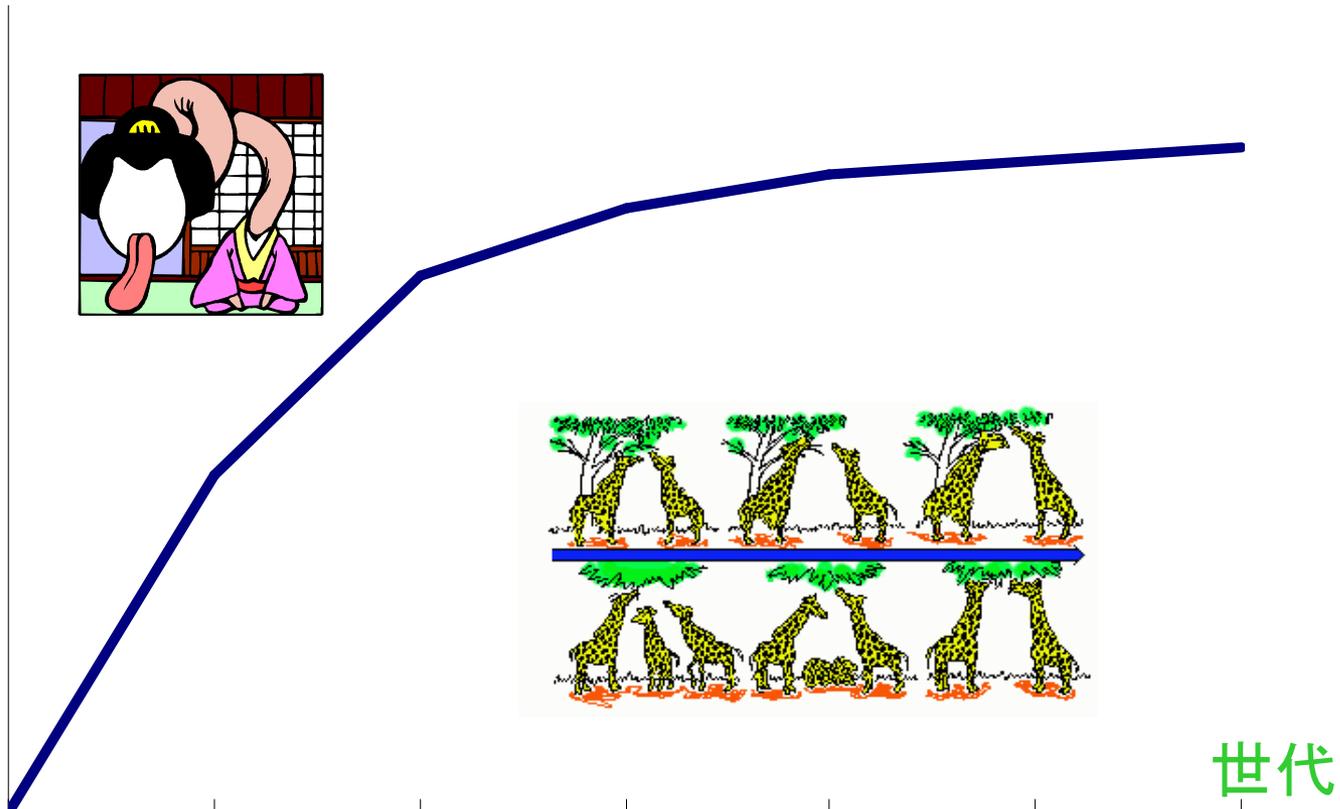


# 進化は進歩ではない(1)

理想的な場合

進化の度合い

適合度



# 進化は進歩ではない(2)

進化の度合い

適合度

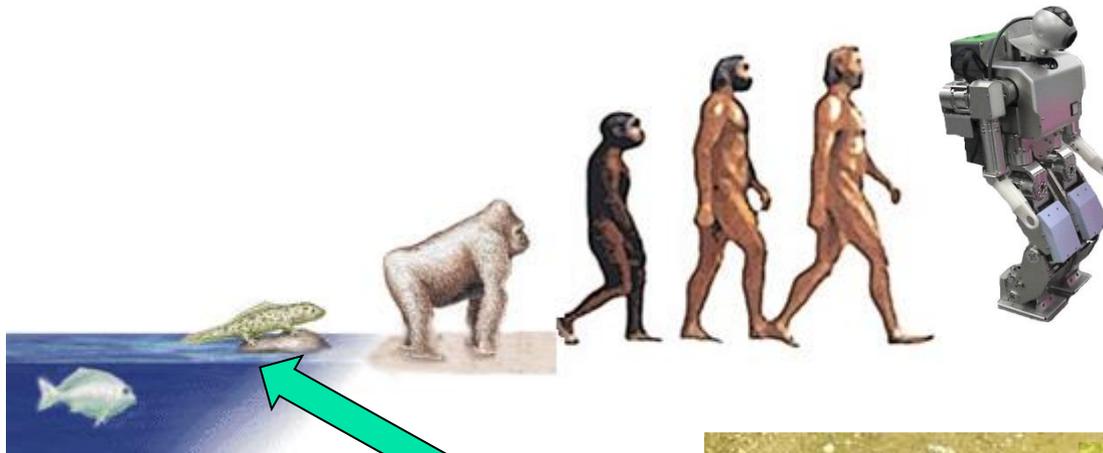
通常の場合



停滞はデータなり

世代

# 間違ったイラスト



- ◆ 最も反映しているのはバクテリアか
- ◆ チンパンジーは何百年後に人間に進化するのか？



ハンドフィッシュ@タスマニア  
魚類から両生類への進化の途上か？

# 算数ができるのは本能か？

- われわれはいつから足し算や引き算などの簡単な計算ができるのか？
- 小学校、幼稚園????
- 実は生後半年にも満たない幼児でも計算の能力があった！！



- Wynn, K. (1992). Addition and subtraction by human infants. *Nature*, 358, 749–750.

# 数字の認知は幼児にもできる

- 同じように $2 - 1 = 1$ のシナリオでも幼児の認知能力は証明されている
- ゾウでも同じようにできるらしい
  - アジアゾウは足し算できるゾウ(入江)
  - <http://beep.c.u-tokyo.ac.jp/~irielephant/>
- 鳥ではどうか？
  - 鳥は本能的に算数はできない??