

# 情報工学実験3:進化計算

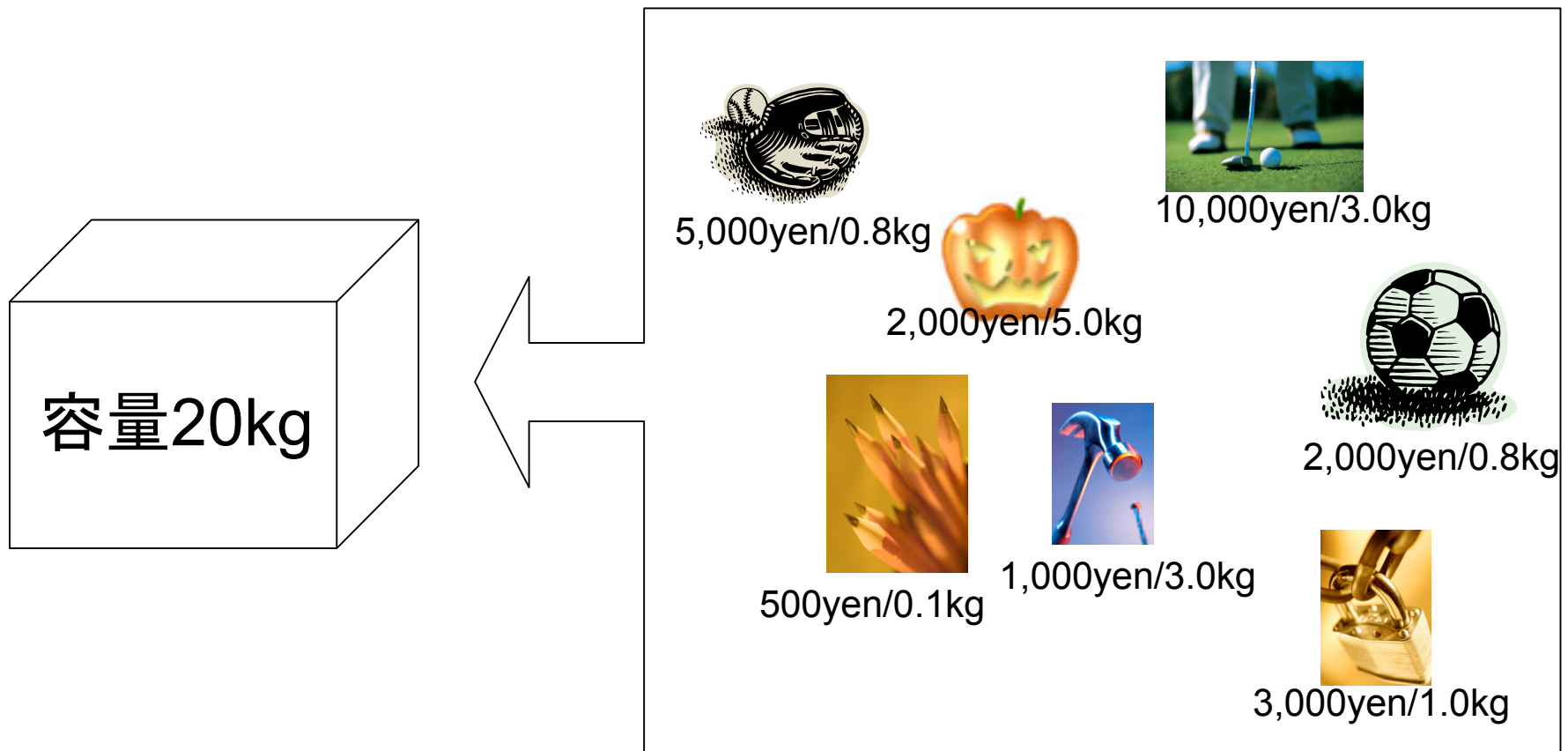
## (week2) GA概要

1. 遺伝的アルゴリズムの用語
2. GAの処理手順 (e.g., ナップサック問題)
  1. コーディング
  2. 適応度評価
  3. 選択
  4. 交叉
  5. 突然変異
3. 復習問題 (e.g., TSP)
4. 設計指針

<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2005/info3/>

# ナップサック問題

- 複数の品物(それぞれの品物は, 重さと値段が異なる)が与えられた時, 重さがナップサックに入る最大重量以内でなるべく合計の値段が最大になるような品物の組み合わせを求めよ.



# ナップサック問題(一般的な記述)

- ナップサックの容積 $V$ のもとで、 $n$ 種類の荷物がそれぞれ異なる大きさ $s$ と価値 $c$ を持ち、それらをナップサックに詰め込むときに、総価値が最大になるような品物の組合せを選択するという問題.

$$\begin{array}{l} \text{目的関数} \\ \text{制約条件} \end{array} \quad \begin{array}{l} \max \{ x_1 c_1 + x_2 c_2 + \cdots + x_n c_n \} \\ V \geq \{ x_1 s_1 + x_2 s_2 + \cdots + x_n s_n \} \end{array}$$

Q: 対象問題は可能な限り抽象化(一般化)した記述が望ましい. それは何故か?  
Q: それを踏まえた上で, 計算実験では何に気をつけるべきか?

# 遺伝的アルゴリズムの用語1

- 個体 (individual)
  - 染色体によって特徴づけられた個
- 染色体 (chromosome)
  - 遺伝子の集まり
- 遺伝子 (gene)
  - 個体の形質を規定する基本構成要素
- 対立遺伝子 (allele)
  - 遺伝子が取りうる値
- 遺伝子座 (locus)
  - 染色体上の遺伝子の位置
- 遺伝子型 (genotype)
  - 染色体の内部表現 (文字列)
- 表現型 (phenotype)
  - 染色体の外部表現 (様々)
- 個体群 or 集団 (population)
  - 個体の集まり

**表現型**  
荷物1, 4, 5, 6を入れる。

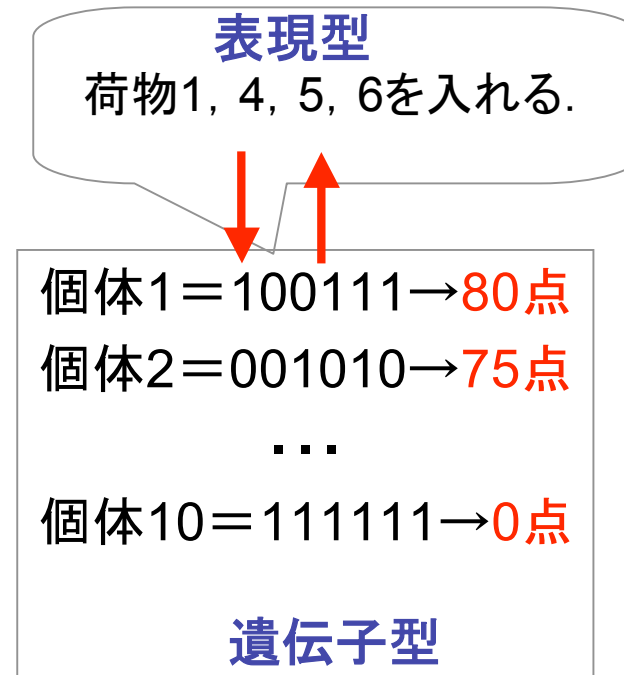
個体1 = 100111  
個体2 = 001010  
...  
個体10 = 111111

**遺伝子型**

Q: どのような遺伝子型が考えられるか?

# 遺伝的アルゴリズムの用語2

- 適応度 (fitness)
  - 個体の環境に対する適合度合い
- コード化 (coding)
  - 表現型 → 遺伝子型
- デコード化 (decoding)
  - 遺伝子型 → 表現型

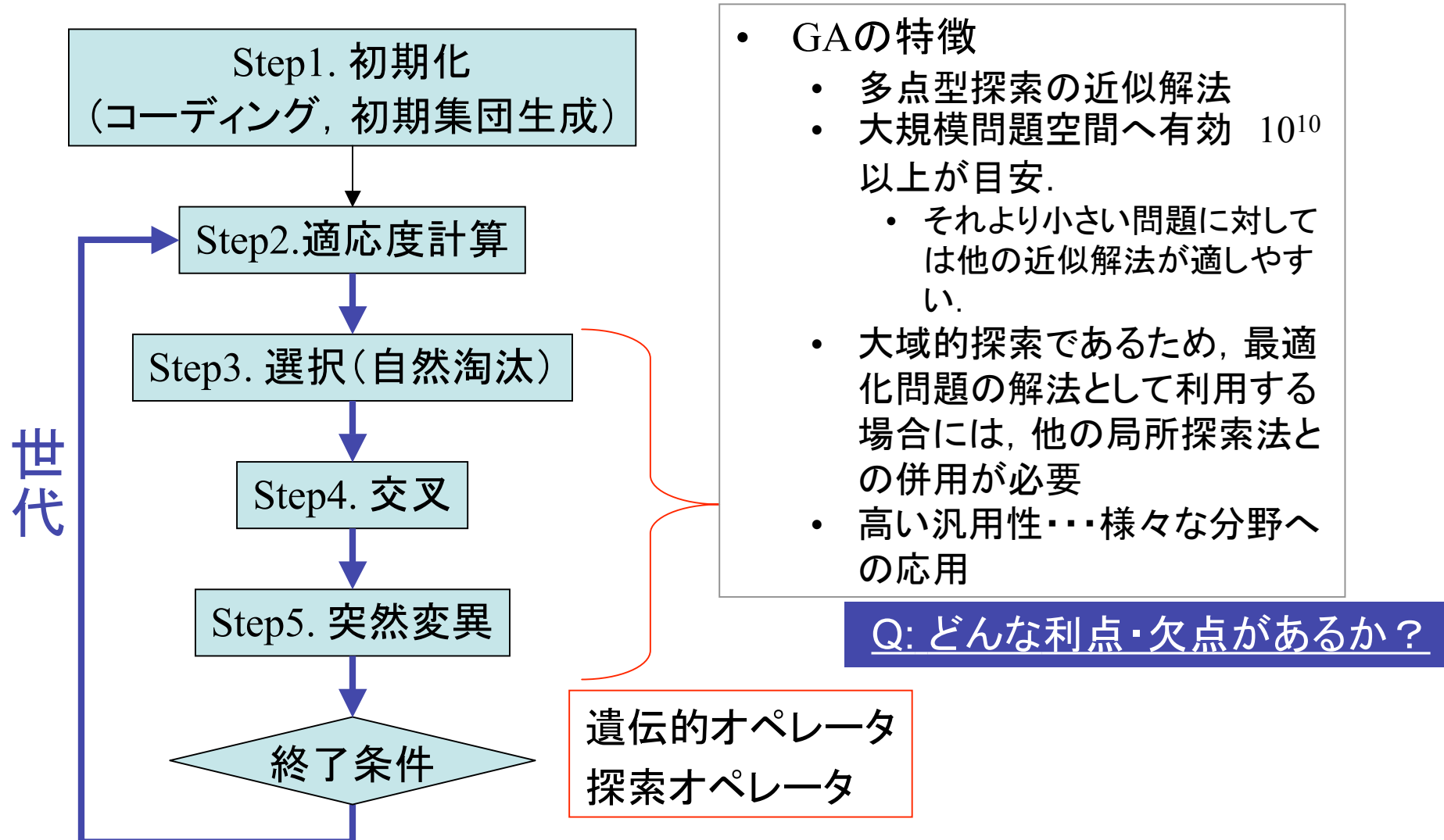


[基本アイデア]

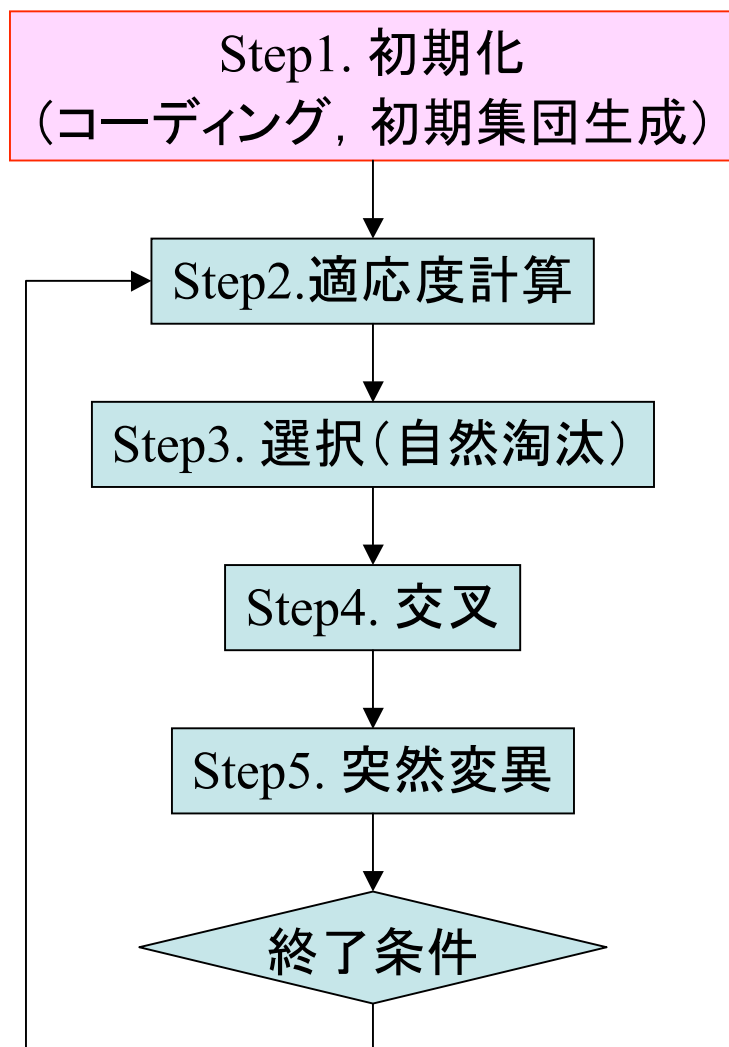
良質の解を組み合わせてより良い解を生成

Q: 確実に良くなるのか？

# GAの処理手順



# Step1. 初期化(コーディング, 初期集団生成)



- コーディング
  - 解の候補を文字列(染色体)として定義
  - 第*i*番目の荷物を入れる→1
  - // 荷物を入れない→0
- 初期集団生成
  - N個体生成
  - ランダム/ヒューリスティック

個体1 = 100111

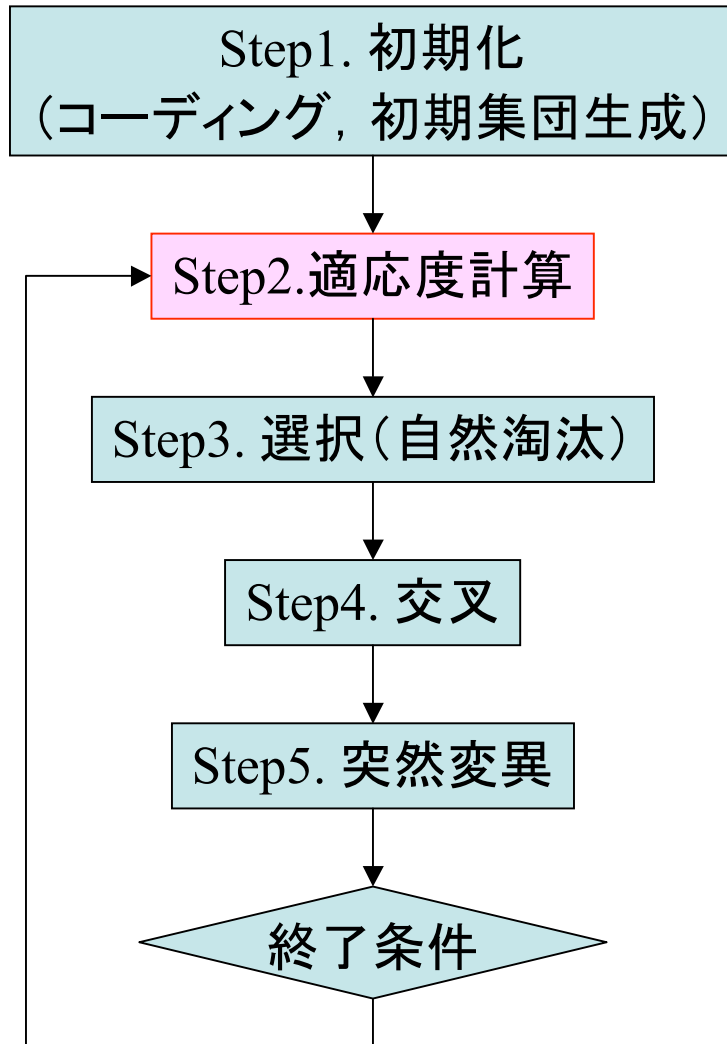
個体2 = 001010

...

個体N = 111111

Q: どんなコーディング・生成法が有効か?

# Step2. 適応度計算



- 環境への適応度を計算
- 処理が重い事が多い
  - 軽い例
    - 数式: ナップサック問題, TSP
  - 重い例
    - シミュレーション: 行動獲得, マルチエージェントシミュレーション, ,

個体1 = 100111 → 80点  
個体2 = 001010 → 75点  
...  
個体10 = 111111 → 0点

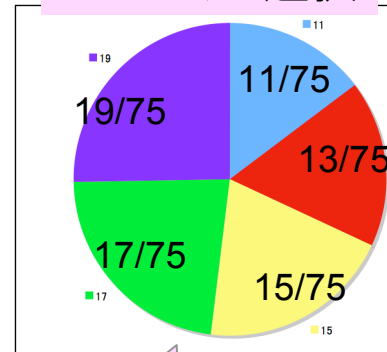


# Step3. 選択(自然淘汰)

- 適応度に依存した方法で選択を行う。
  - 適応度の低い個体は淘汰
  - 高い個体は増殖
- ルーレット選択(適応度比例選択)
  - 個体群の中の各個体の適応度とその総計を求めて、適応度の総計に対する各個体の割合を選択確率として個体を選択する.
- ランキング選択
  - 適応度の大きなものから順に次世代に残す個体数を予め決めておく(選択確率固定)という方法.
- エリート保存選択
  - 個体群の中で最も適応度の高い個体は無条件でそのまま次世代に残すという選択方法.

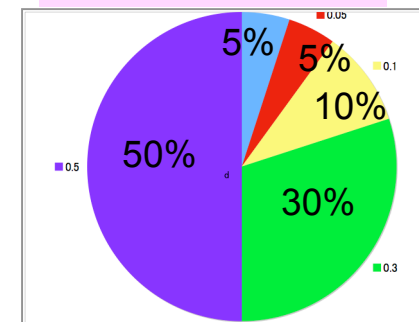
e.g.) S1:11点, S2:13点, S3:15点, S4:17点, S5:19点

ルーレット選択



適応度が高い解ほど残りやすい

ランキング選択

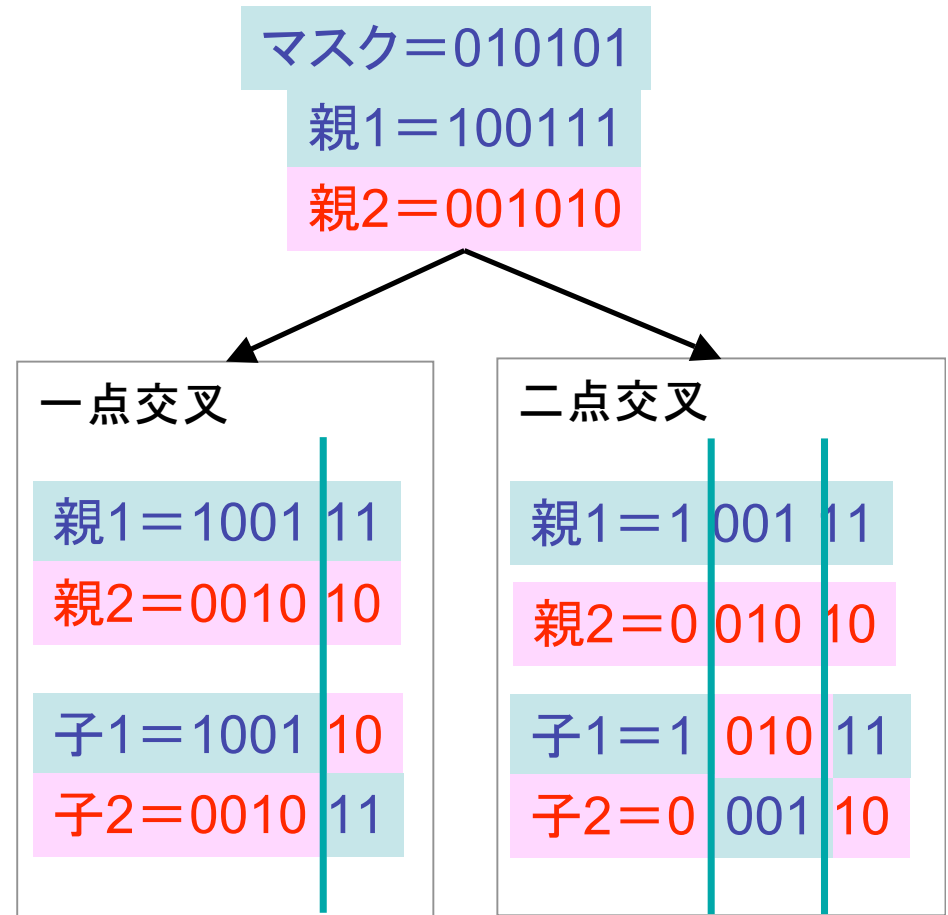


適応度の差異が少ない場合でも、高い解を残しやすい

高い解は必ず残る

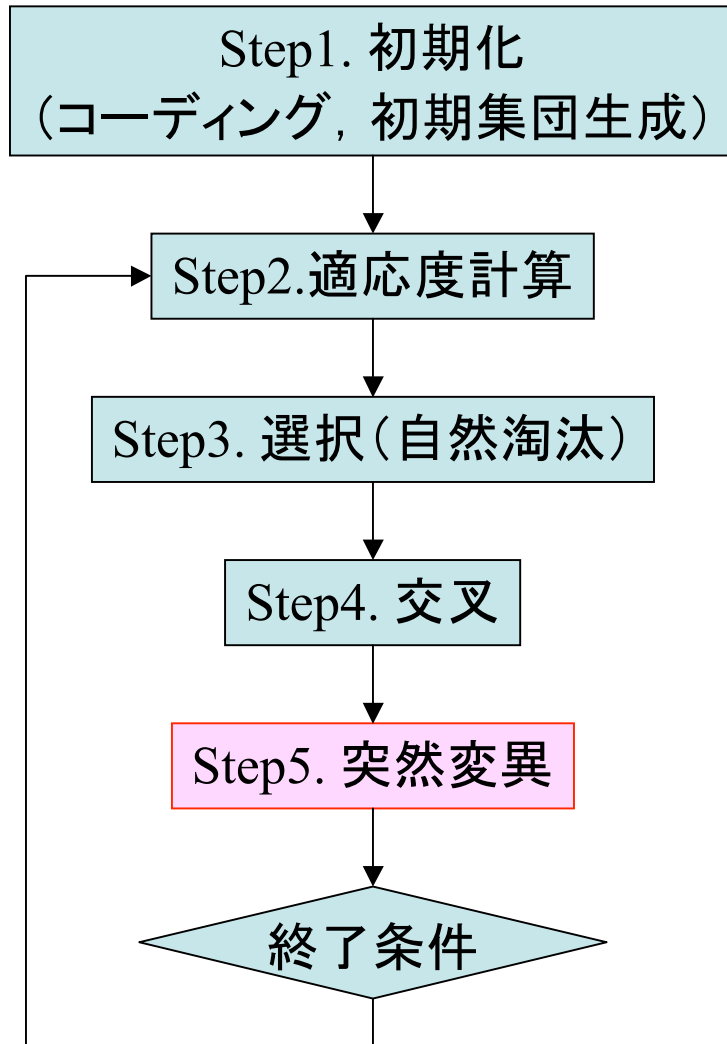
# Step4. 交叉

- ランダムに親個体を2個体選択し, その2個体の掛け合わせにより新しい子個体を生成.
- 交叉確率
- 一点交叉
  - 切断箇所1点
- 二点交叉
  - 切断箇所2点
- 一様交叉
  - マスク処理

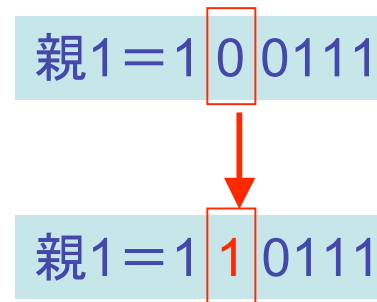


Comment: 致死遺伝子との  
兼ね合いに留意

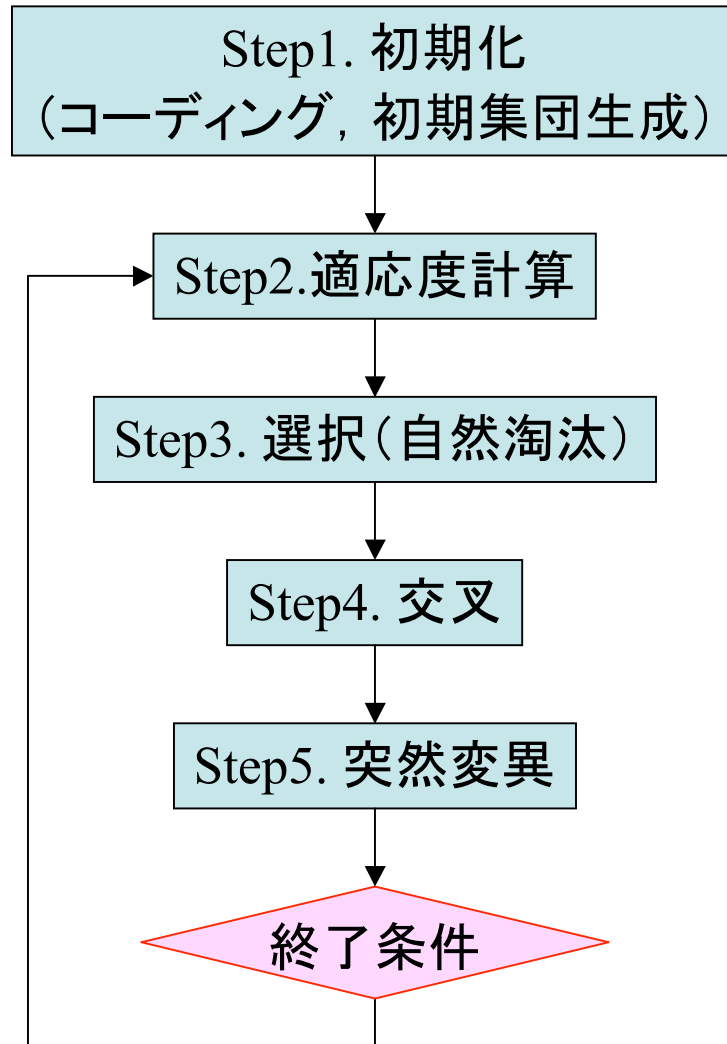
# Step5. 突然変異



- 染色体上のある遺伝子を一定の突然変異確率で他の対立遺伝子に置き換える操作.

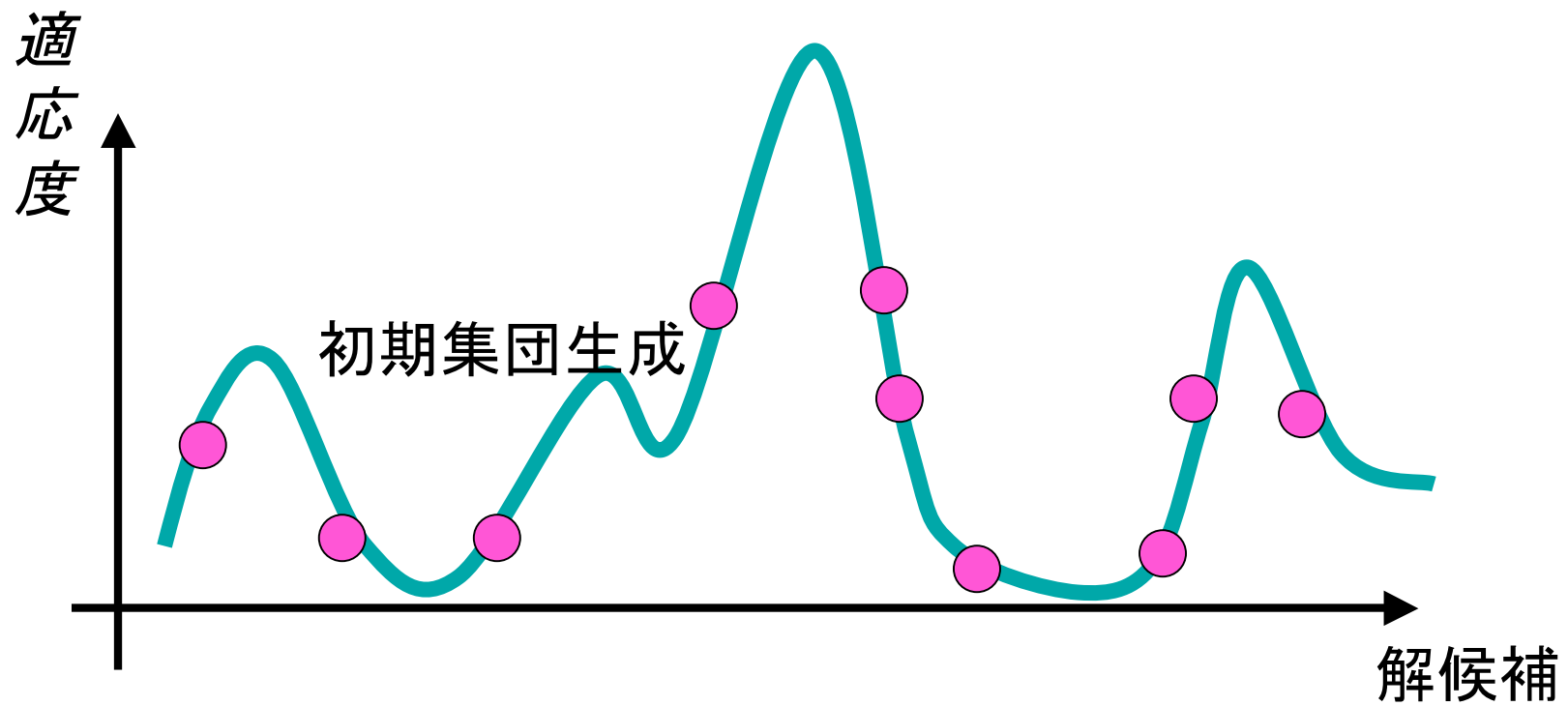


# 終了条件

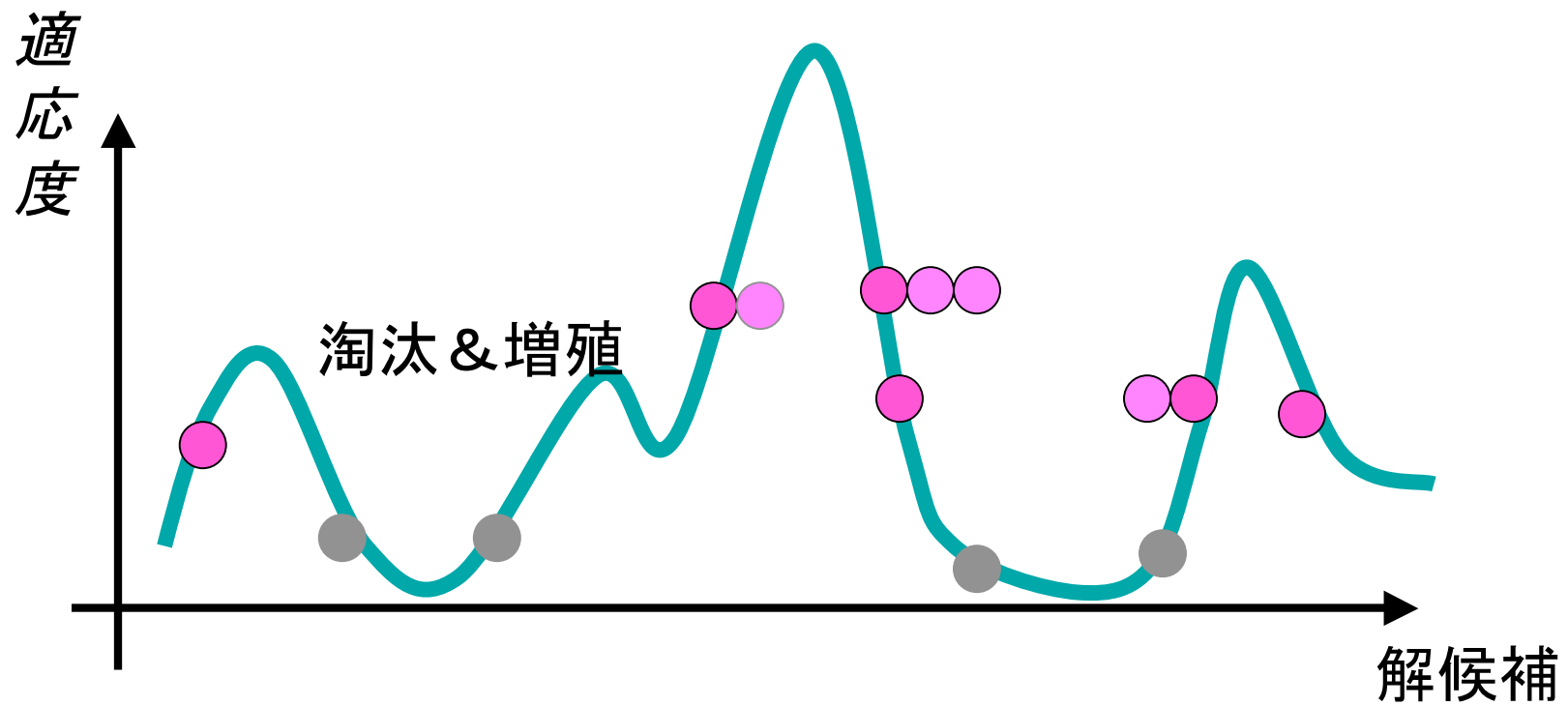


- 個体群の中の最大の適応度が、設定されたしきい値を超えたか.
  - 個体群全体の平均適応度が、設定されたしきい値を超えたか.
  - 世代交代の回数が、あらかじめ設定した回数を超えたか.
- など.

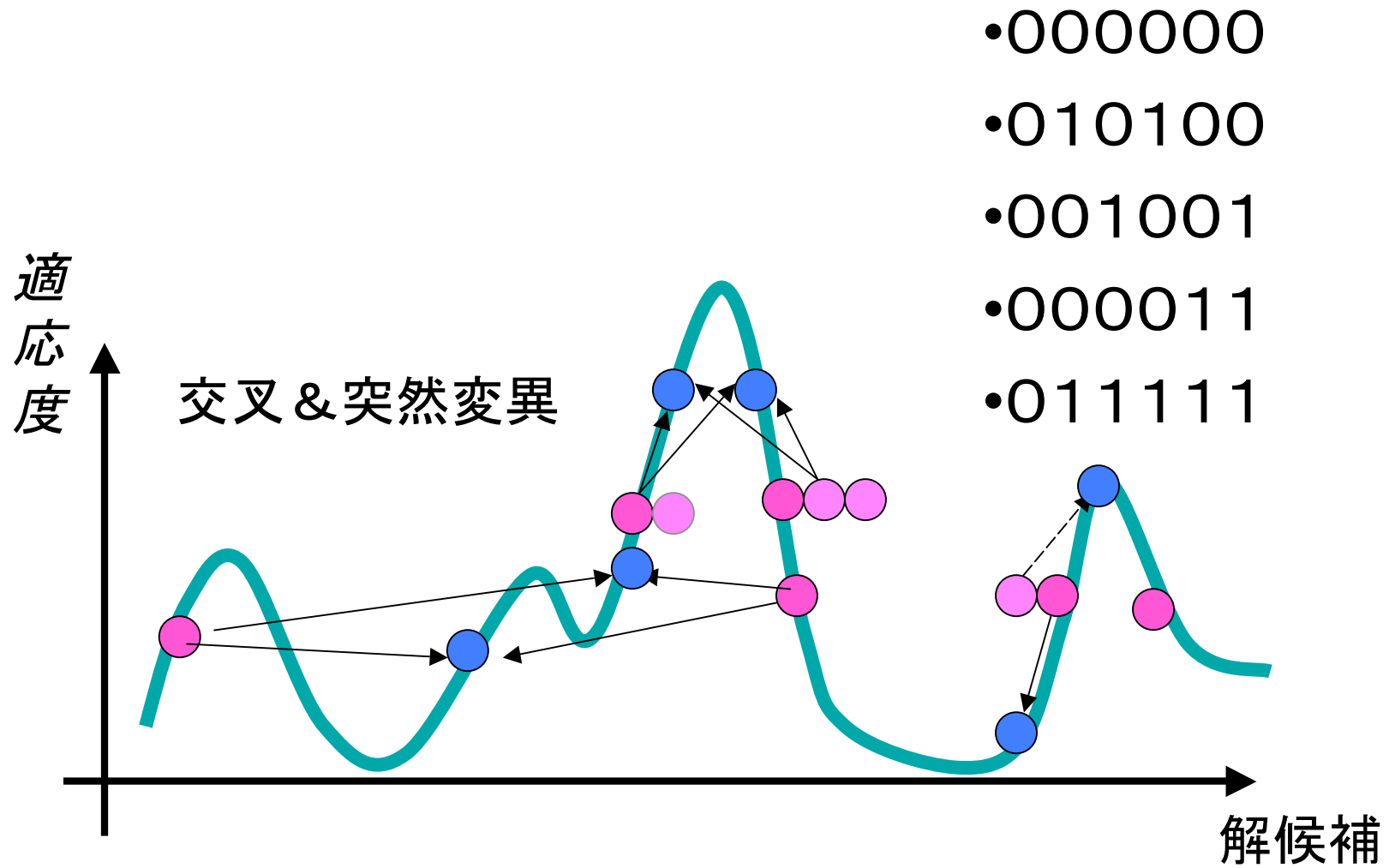
# GAは何をやっているか？



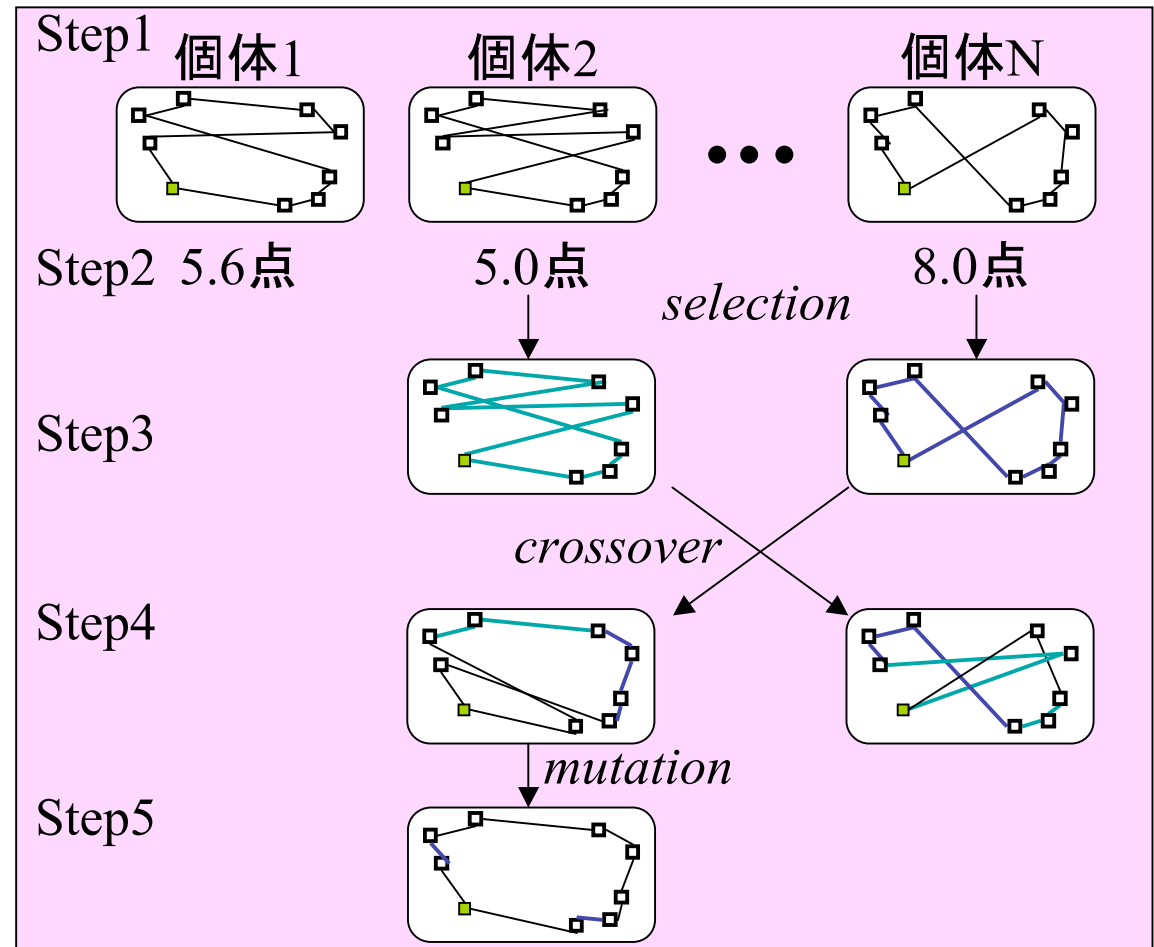
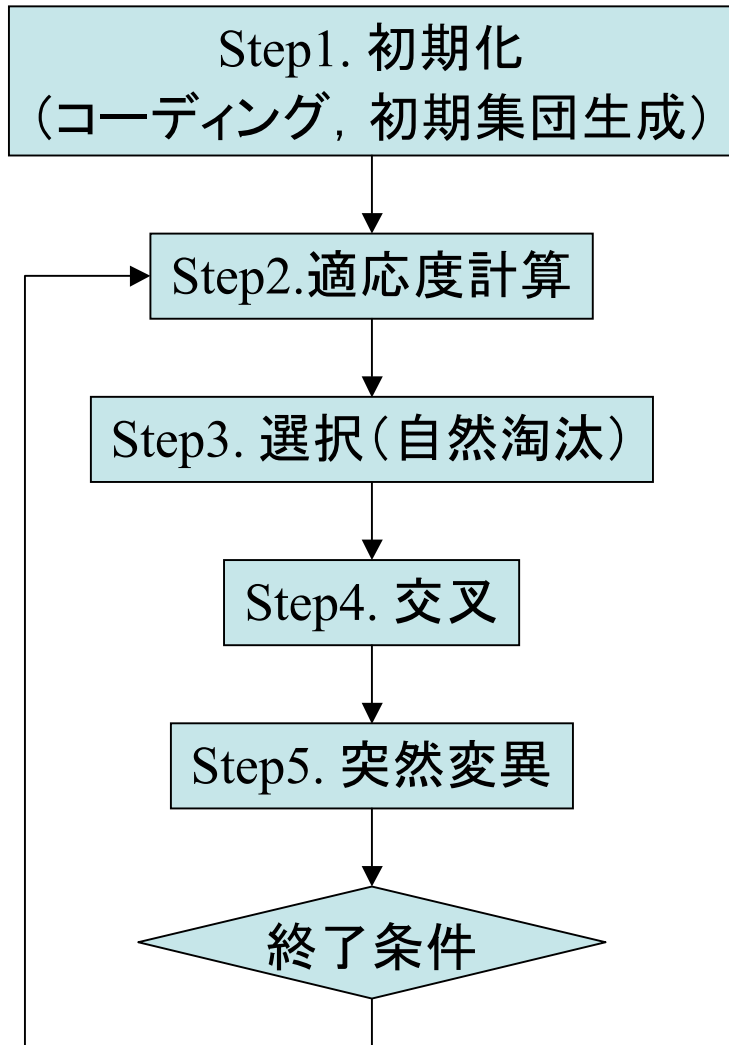
# GAは何をやっているか？



# GAは何をやっているか？



# 復習問題:TSP





# 設計指針とTSPにおける例

## (設計1) コーディング設計

解  $x$  をどのように表現するか

巡回する都市番号を巡回順番に並べて表現.

解  $x =$ 

1	7	6	4	5	8	9	3	2
---	---	---	---	---	---	---	---	---

## (設計2) 目的関数設計

解に対する評価をどう設定するか

巡回に要する巡回経路長を計算.

$$f(x) = \text{length}(x)$$

## (設計3) 探索オペレータ,

### パラメータ設計

選択・交叉・突然変異等

- ルーレット選択 + エリート保存戦略
- サブツアー交換交叉
- 点突然変異

# まとめ

- 問題の定式化
    - ナップサック問題, TSP
  - GA
    - Step1. 初期化
    - Step2. 適応度計算
    - Step3. 選択
    - Step4. 交叉
    - Step5. 突然変異
  - 設計例(TSP)
- 課題: Level2
    - GAの特徴を3つ以上挙げ, その利点・欠点について考察せよ. 他の探索アルゴリズムと比較するとなお良い(下記オプション).
  - (オプション)
    - 時間があれば, さらに他の最適化手法(全探索・山登り法・ニューラルネット・タブー探索等何でも可)についても同様に検討し, 手法としての違いについて指摘せよ(加点対象). (例えば, 具体的な対象問題について適用しやすさ・手法選択の妥当性等の観点から指摘する等)

# 次回

- (必要に応じて)GA復習
- GA関連研究の事例紹介
  - GAそのもの(未定)
  - インタラクティブGA
  - 遺伝的プログラミング(GP)