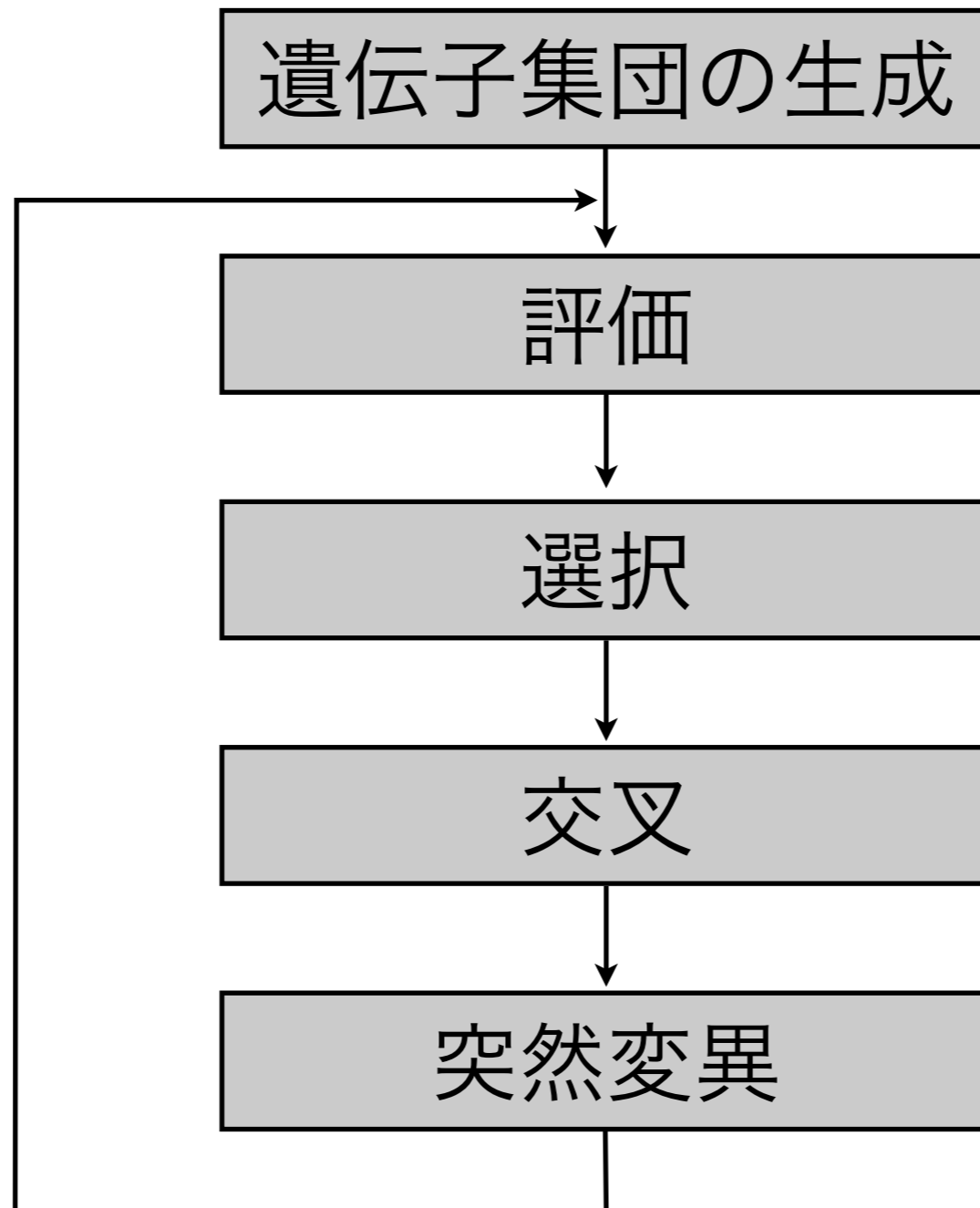
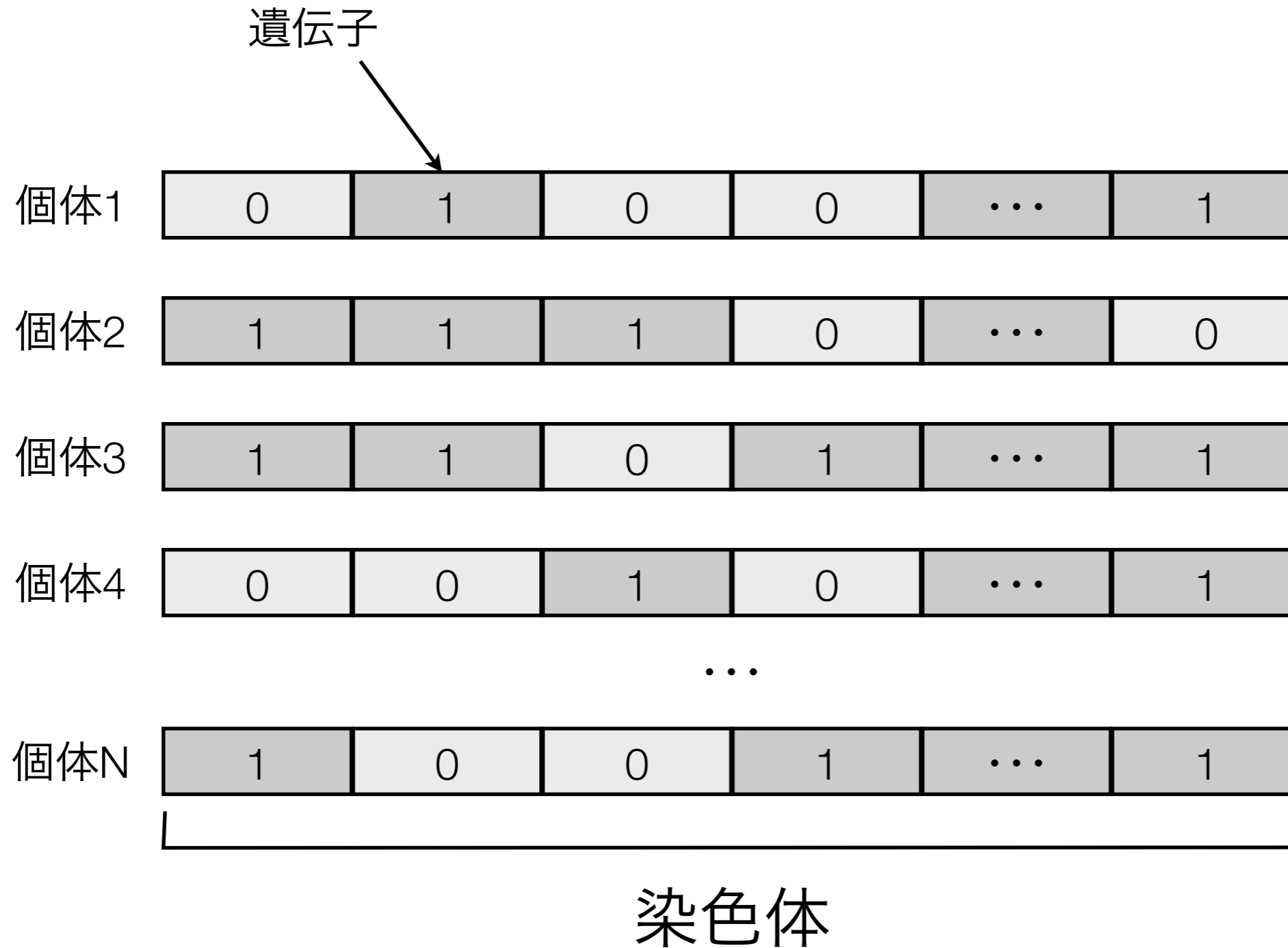


遺伝的アルゴリズム

遺伝的アルゴリズムの流れ



遺伝子集団の生成



評価 (fitness evaluation)

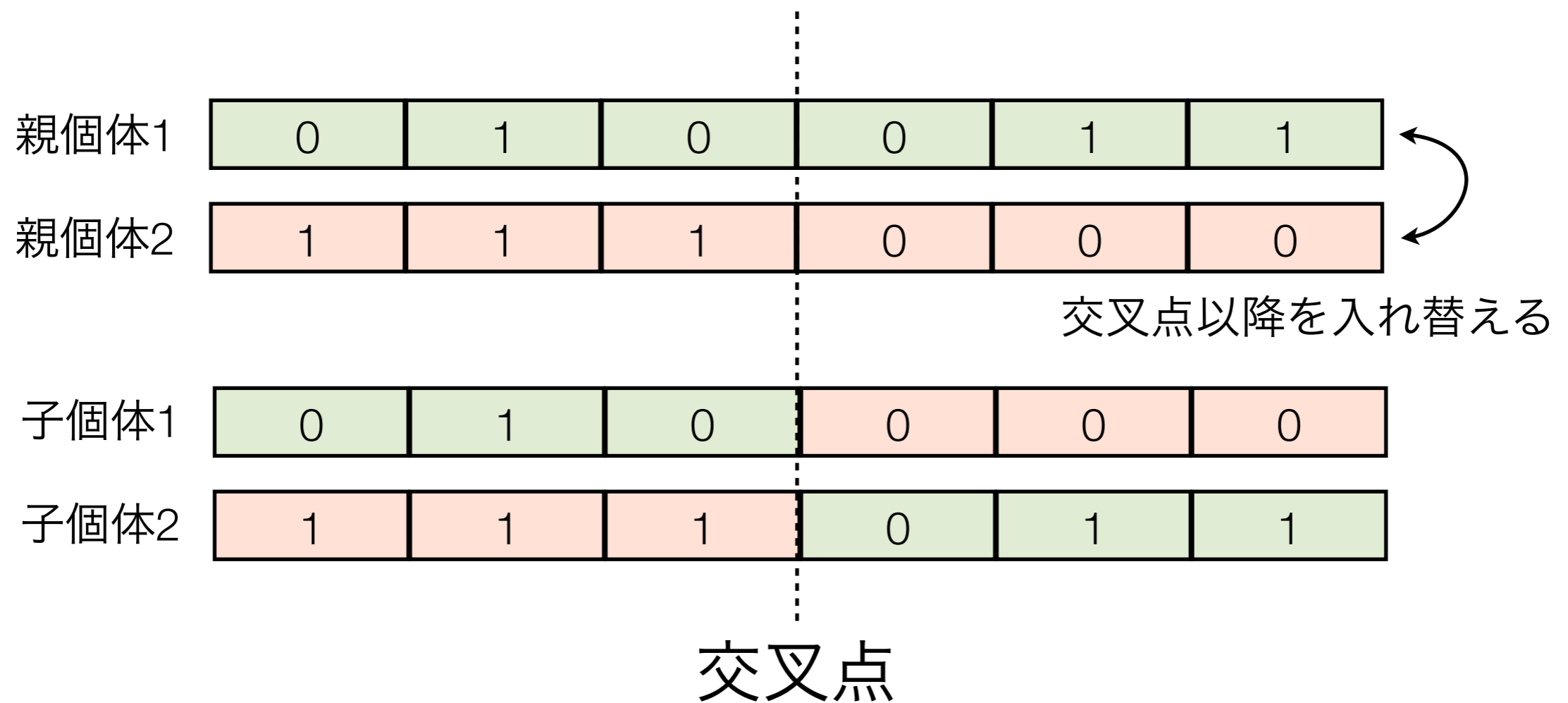
- 復号：染色体（コンピュータで操作可能な数列） → 表現型
- 表現型 = 解候補 ← 評価できる
例) 数値, 組み合わせなどなど
- 評価値：個体の適応度を表す数値
(スカラーでなくベクトルを扱うものなどもある)

選択・淘汰 (selection)

- 次世代の遺伝子集団を生成する → 初期収束の防止 = 多様性の維持が重要
- 評価値 (適応度) の高い個体ほど次世代に生き残りやすい
- 選択方式
 - ルーレット選択 (最も一般的)
 - ランク方式
 - トーナメント方式
- サプリメント
 - エリート戦略: 最も良い個体は優先的に残す

交叉 (crossover)

- 2つ以上の染色体を混ぜ合わせる (一点交叉, 多点交叉, 一様交叉など)
- 交叉率 (crossover rate) で決められた数の親個体を子個体で置き換える



突然変異 (mutation)

- 各遺伝子毎に突然変異率 (mutation rate) に従って遺伝子の値が変わる
 - 2値遺伝子の場合は $0 \rightarrow 1$ または $1 \rightarrow 0$
- 突然変異率は通常かなり低い値
 - 経験的には $1/\text{遺伝子長}$ (1個体に含まれる遺伝子の数)

アルゴリズムの終了条件

- いろいろな終了条件がある
 - 個体集団に含まれる染色体の多様性が一定以下となったとき
 - 規定回数世代を繰り返したとき
 - 解候補が十分に良くなったとき

関数の解を求める (ga.solver.equation)

- 数式 $f(x)$ の値が最大となるような x を求める
- x を遺伝子に符号化 → 2進数的に符号化
(ga.solver.equation.Problem.java : decodeメソッド)
- x を式に代入し出てきた値を (ほぼ) そのまま評価値とする
(ga.solver.equation.Problem.java : evaluateメソッド)

ナップサック問題 (ga.solver.knapsack)

- ナップサックに入れる荷物の価値の合計を最大にしたい
制約) ナップサックに入れられる荷物の重さには限度がある
- ナップサックに入れる荷物の価値の合計を評価値
ただし, 重さの合計が限度を超えた場合は評価値=0
- 符号化: 遺伝子の数=荷物の数
→ ナップサックに入れる荷物は 1 入れない荷物は 0