メタ最適化とは

最適化工学技塾

清水　良明

メタヒューリスティック解法

複雑な対象(シミュレーションベース)
大局的最適化(多峰性)
組合せ(離散的)問題

- GA (Genetic Algorithm)
- SA (Simulated Anealing)
- TS (Tabu Search)
- DE (Differential Evolution)
- PSO (Particle Swarm Optimization)
- Memetic Algorithm
- ACO (Ant Colony Optimization)
- Scatter Search

メタ解法のイメージ

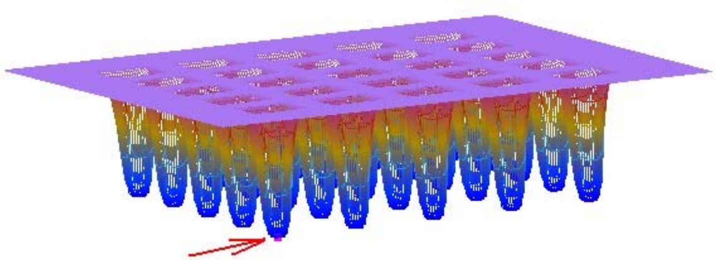

どのようにしてどこから
探せばよいのかな？
最適点をいついつまでに探して下さい
1. まず回りを探してみて下さい。
2. よい点が見つかりましたか。
3. そこまで進んで下さい。
4. 探す上で気付いたことを参考にして、
次の探索範囲と探し方を工夫して下さい。
5. 満足いくまで続けて下さい。
6. 約束の時間がたちましたが
どうですか。
メタ解法
1. はい、わかりました
2. はい、ここです。
3. はい、着きました。
4. はい、わかりました。
範囲はこのようにします。
探し方はこのようにします。
5. はい、わかりました。
6. この点が最も秀れています。
これで満足です。

多峰性関数の最適化

Schekel’s function (fox hole)

$$f(x) = 1/(0.02 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6})$$

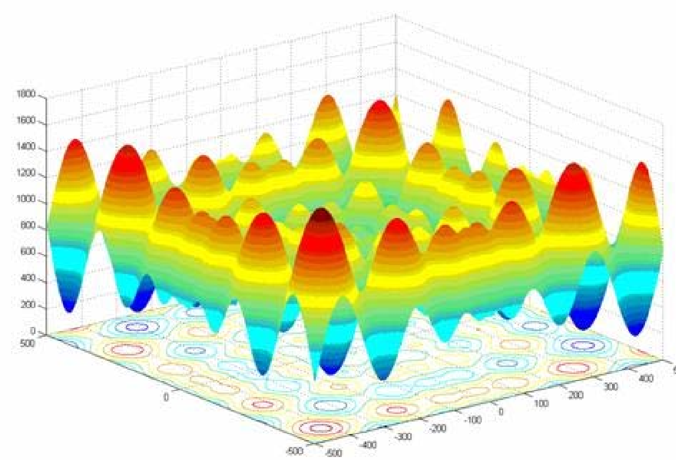
$$-65.535 \le x_i \le 65.536$$



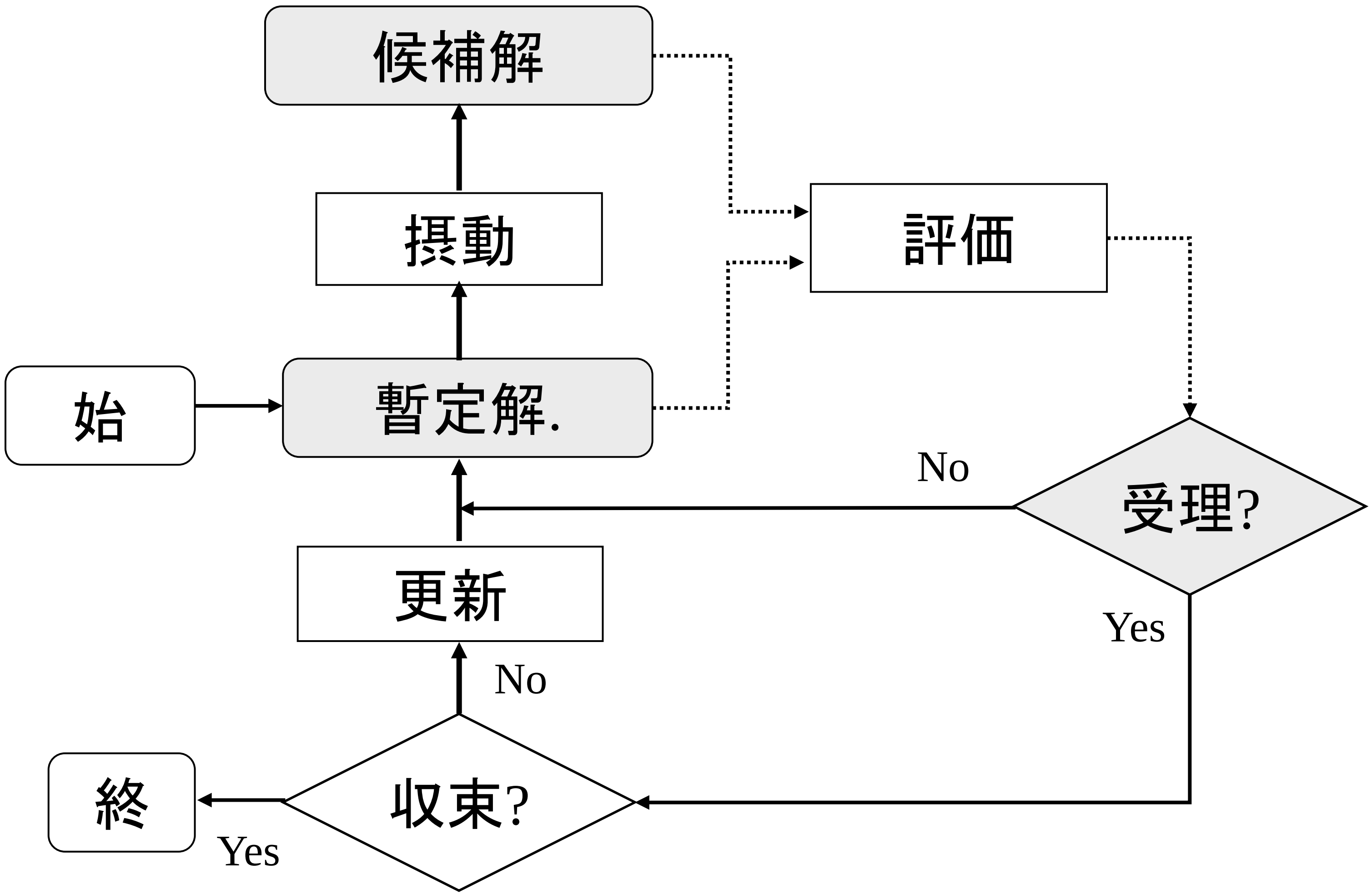
Schewelfel’s function

$$f(x) = -\sum_{i=1}^{2} x_i \sin(\sqrt{|x_i|})$$

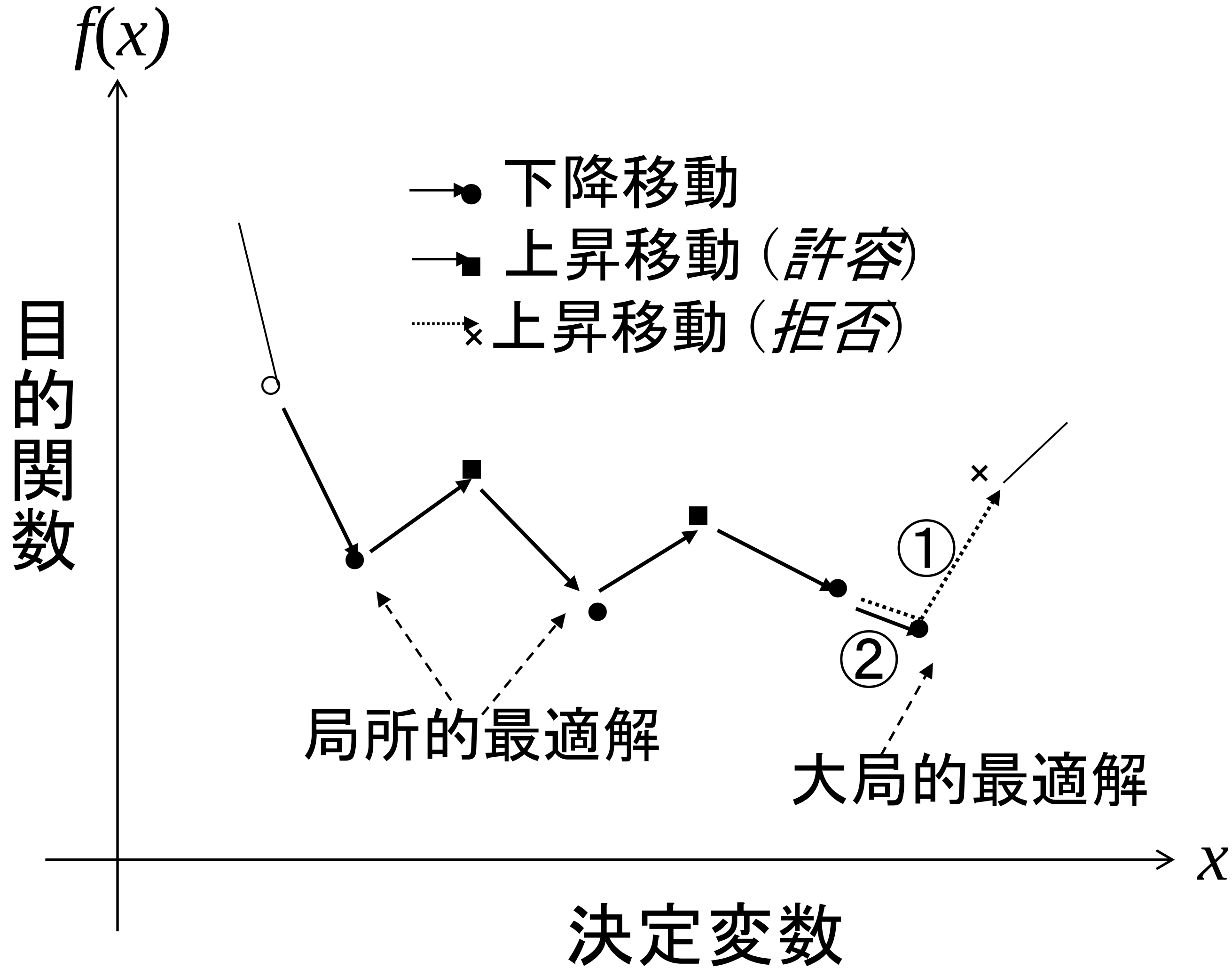
$$-500 \le x_i \le 500$$



メタ解法の一般的流れ



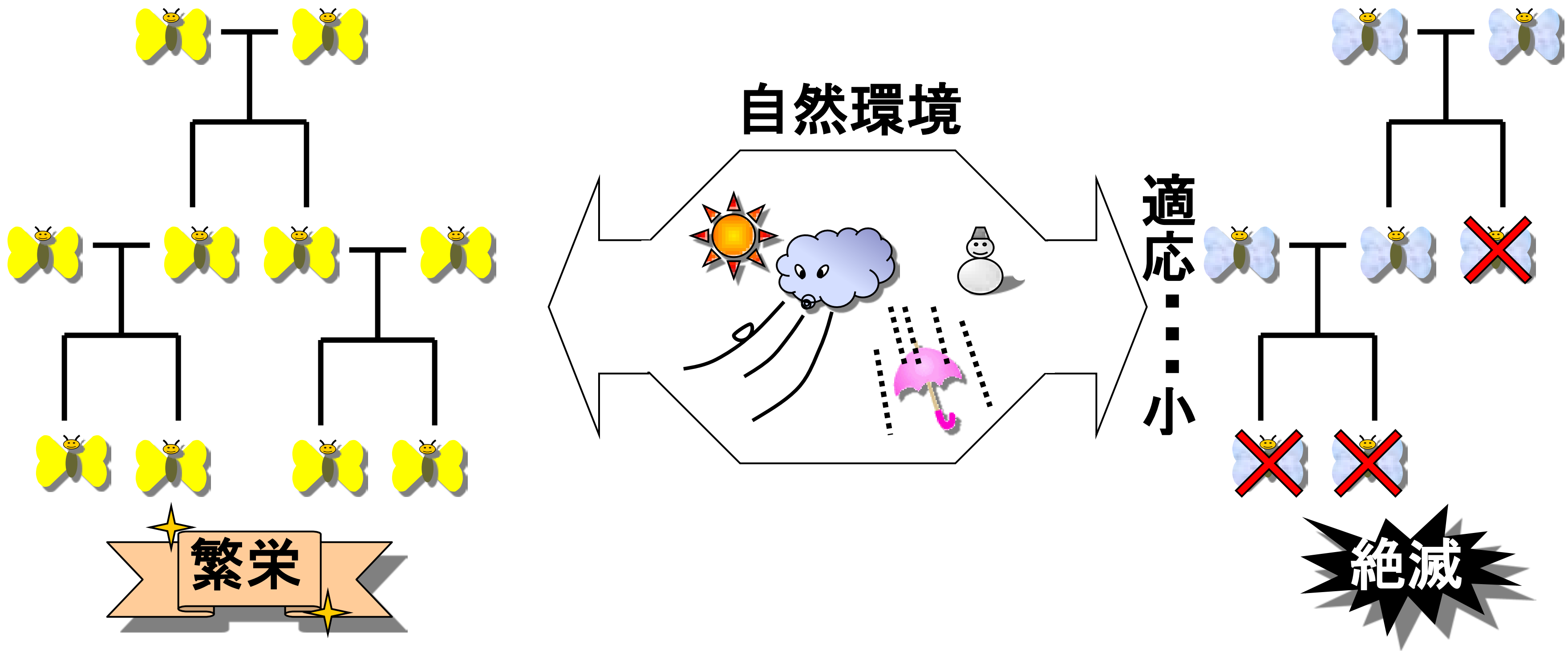
メタ解法の大局最適化原理



4. 遺伝アルゴリズム(GA)の概要

“自然界のシステムの適応過程における生物進化のメカニズムを人工的に模擬する手法“ (Hollandらによって1960年代に提唱)

「自然の進化では、生物同士が交配と突然変異を繰り返しながら、
自然環境によく適応する生物集団ほど生き残り、
子孫を増やすことができ、逆に適応しなければ死滅する。」



最適化問題の求解アルゴリズムへの応用

$$(p.1)\quad Min\quad f(x)\quad subject\quad to\quad x \in X$$

X : 決定変数 $\mathbf{x}$ の実行可能領域

(1) 決定変数 x に染色体を対応させ記号列で表す。

$$x:\ A_1 A_2 \cdots A_{\mathrm{i}} \cdots A_{\mathrm{n}} \qquad (1)$$

遺伝子: 記号 A_i $(i = 1, \ldots, M)$

遺伝子座: 遺伝子が置かれる位置　$(i = 1, \ldots, M)$

対立遺伝子: 遺伝子が取り得る値　(e.g. 0 か 1)

(2) 最適化の t 回探索後の解集合を t 世代進化後の個体群に対比

$$\mathrm{X}(t)=\{x_1(t), x_2(t), \cdots, x_k(t)\}, \quad k:\ \text{個体群のサイズ} \qquad (2)$$

(3) 探索アルゴリズムを進化則(生存度は適応度に応じる)に対比
(最適化問題の目的関数値を用いて適応度を定義)

（例1） 0から1の間にある最適解を0.02以下の精度で求めたい。

このときバイナリーコーディングで必要となる染色体の長さを求めよ。

ただし、2^4＝16 、2^5＝32、2^6＝64、2^7＝128である。

2^0	2^1	････	2^{r-1}

r ビットで表される最大の数＝2^r－1

制約条件付き問題の取り扱い

いずれかの制約式を満たさない個体は
淘汰されるように適応度を定義し直す。

$$X = \left\{ x \,\middle|\, \begin{array}{l} g_i(x) \le 0 \quad (i = 1, \ldots, L) \\ h_i(x) = 0 \quad (i = 1, \ldots, M) \end{array} \right\}$$

<u>制約条件付き問題を制約条件無し問題に変換</u>

$$f'(x) = f(x) - P\{\sum_{i=1}^{L} Max[0, g_i(x)] + \sum_{i=1}^{M} h_i^2(x)\}$$

ここでPは、ペナルティ係数で正の大数

アルゴリズムの実現

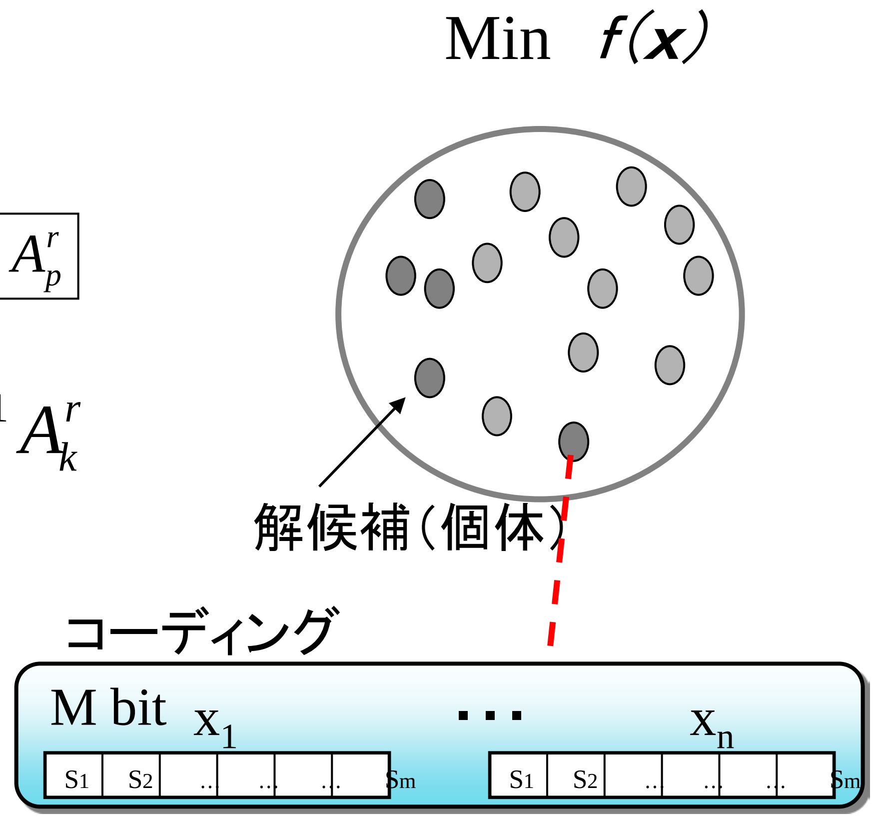
遺伝子の表現型:

l_kの組を個体群に対応させる。

$$\underbrace{\boxed{A_1^1 A_2^1 \cdots A_p^1 \mid A_1^2 A_2^2 \cdots A_p^2}}_{2^0 \qquad 2^1} \cdots \underbrace{\boxed{A_1^r A_2^r \cdots A_p^r}}_{2^{r\text{-}1}}$$

$$l_k = A_k^1 + 2^1 A_k^2 + 2^2 A_k^3 + \ldots + 2^{r-1} A_k^r$$

r: 表現単位の繰り返し数
対立遺伝子: {0, 1}
等分数、$L = 2^r\text{-}1$



適応度:

$$F_i(t) = \{S^T(i, t) \,/\, \underline{S}^T(t)\ \text{-}1\}^3$$

$\underline{S}^T(t)$: t 世代の個体群中の最小の総合得点

終了条件:

(1) 一定世代経過後の一定期間、群全体として適応度の改善がない
(2) 最良の個体群の増殖が停止したとき

スケーリング

適応度の差異を適切に拡大
　　あるいは縮小させる方がより効果的となる

(1) 線形スケーリング(Linear scaling)

$$F_i' = aF_i + b \qquad (4.6)$$

(2) シグマ切断(Sigma truncation)

$$F_i' = aF_i - (\overline{F} - c\sigma) \qquad (4.7)$$

ここで $\overline{F}, \sigma$ は、それぞれ個体群の適応度の平均値と標準偏差を、また c は1～3の定数を表わす。

(3) ベキ乗スケーリング(Power law scaling)

$$F_i' = (F_i)^k, \quad (k>1)$$

GAの遺伝演算（淘汰、交叉、突然変異）

1. 淘汰の規則：

適応度の高低に応じて個体を死滅、増殖させる。

(a) ルーレット方式

個体 x_i を、F_i / F_Tの確率で選択 ($F_T = \sum_{k=1}^{NP} F_k$)

F_i: 個体 x_i の適応度 ← 目的関数値 f_i から換算

1. 確率的選択

乱数rand( )を発生させ、$\sum_{i=1}^{k} F_i \geqq \text{rand}(\ )F_T$となるような最小の k を求めて、k 番目の個体を次世代に生き残る個体とする。

この手順を生き残る個体数がN_p となるまで繰り返す。

2. 期待値選択

個体 x_i を選択確率 p_i に対する期待値として選択する。
次世代に淘汰される個体数をN_Dとするとき、個体x_iは$[p_i N_D]$だけ増殖する。
ここで$[\cdot]$はガウス記号

(b) ランキング方式

適応度の非常に高い個体が含まれるような場合、その個体だけが増殖
適応度に差がないと、いつまでたってもよい個体が増えない。

- 数値よりも順序を重視する。
- 適応度に基づいて各個体をあらかじめ第１位から最下位までのランクづけ
- 各個体のランク i, i=1, 2, ･･･ 順にあらかじめ選択確率を定めておき、それに基づいて個体の選択を行うもので、バランスよい再生が期待できる。

第 i ランクの個体を

- 線形ランキング選択　：　$p_i=a+b(i-1)$
- 非線形ランキング選択　：　$p_i=c(1-c)^{i-1}$

で再生する。

(c) トーナメント選択

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、
その中で最も適応度の高い個体を(トーナメント方式で)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。

(d) エリート保存方式

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。
最良の個体は、交叉や突然変異により破壊されることはない。
局所的な解に陥る危険も含んでいる。
他の選択手法と組み合わせて使用される。
個体群の中の適応度の最大値は単調増加する。

2. 交叉の規則

最も重要な役割を果たす演算子

個体群よりランダムにペアを作り、一定の確率P_cで交叉させる。

1点交叉 : 染色体上でランダムに切れ目を一つ選び、
切れ目の右側の部分列を相互に入れ換える。

（例） 親1 01001 | 101 ➡ 子1 01001 | 110
親2 01100 | 110 子2 01100 | 101
切れ目 （対立遺伝子が{0, 1}の場合）

2点交叉 : 染色体上でランダムに切れ目を2つ選び、
切れ目の両側の部分列を相互に入れ換える。

（例） 親1 01 | 001 | 101 ➡ 子1 01 | 100 | 101
親2 01 | 100 | 110 子2 01 | 001 | 110

(c) 一様交叉(uniform crossover): {0, 1}を等確率で発生させて 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の形質を、0の遺伝子座には親2の形質を受け継ぐ子1と、その逆の受け継ぎ方をする子2を作る。

例えば、マスクパターンを 01101101 とすると以下のようになる。

親1: 01001101 → 子1: 01001111
親2: 01100110 → 子2: 01100100

交叉のアルゴリズム

① N_p個の個体からなる個体群の中からランダムに2つの個体(親)を取り出す。k =1として、次のステップへ進む。

② 親の個体間で適当な交叉を行い、生成された新しい2つの個体(子)をもとの個体群に戻す。$k = k+1$として、次のステップへ進む。

③ $k > [P_c N_p]$ならば終了する。そうでなければステップ①へ戻る。

3. 突然変異の規則

“交叉の役割を補い、個体群の多様性を維持する働き
確率P_mで、ランダムに選ばれた遺伝子座の遺伝子を
他の対立遺伝子に置き換える。“

（例）　第3遺伝子座が選ばれるとすると

01101101　→　01001101

他にも次のような操作がある。

(a) 転座: 染色体の一部分が同じ染色体の他の部分
または他の染色体上に位置を変える。
(b) 重複: 染色体上である長さのコードを重複させる。
(c) 逆位(あるいは反転): 染色体上で
部分的に遺伝子の配列順序を 入れ換える。
(d)挿入: 染色体上である長さのコードが挿入される。
その結果、染色体の長さが変わる
(e) 欠失: 染色体上である長さが失われる。
この場合も、染色体の長さは変わる。

GAの一般的な探索過程

Step 1. 世代を $t = 0$ とする。ランダムに初期個体群 $X(0)$ の設定

Step 2. 各個体の適応度に応じ、適応度の低い個体を死滅させ、適応度の高い個体を増殖させる。

Step 3. 一定の確率で交叉や突然変異を行い、

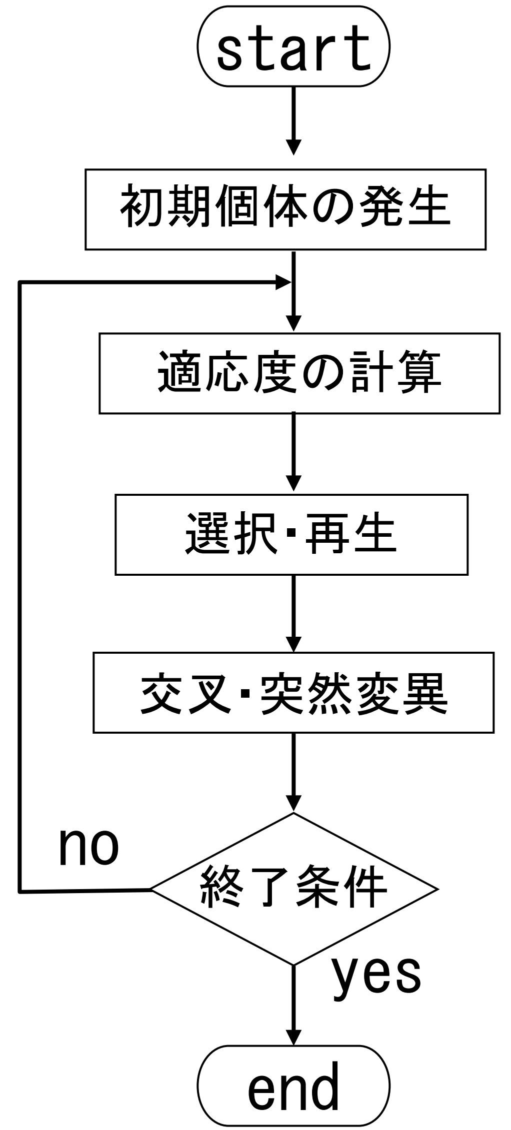
新しい個体(子)を生成する。

子はその生成に関与した古い個体(親)と置き換わる。

(Step 2-3から新しい世代の個体群 $X(t+1)$ が生成される。)

Step 4. 終了条件が満たされた時点での最良の個体を(準)最適解として終了する。

$t = t+1$としてStep 2へ戻る。



その他の知見

(a) 初期個体群を設定するとき、必要な個数の数倍αN (ただし$\alpha>1$)の個体をランダムに生成して、その中から適応度の高いN個を選び出す。

(b) 終了条件としては、あらかじめ繰り返し回数を定めておいたり、適応度の改善が飽和した場合に終了することが多い。

(c) 突然変異は優れた個体を破壊する。$P_c > P_m$に設定する。P_mは世代が進につれて小さくする。

GAの特徴

1. 十分世代を経れば最適解に収束する。
2. ランダム法に比べて効率的
3. 多峰性のある最適化問題の解法としても有効（一点より多点からの探索）
4. 探索手続きが統計的な性格を持つ

（Simple）GAの調整

遺伝子型をどのように表現するか？

に加えて

パラメーターのチューニング

(1) 個体群のサイズ
(2) 世代の終了値
(3) 交叉の確率
(4) 突然変異の確率

GAの適用の成功の鍵

↓

遺伝演算のバランス良い適用

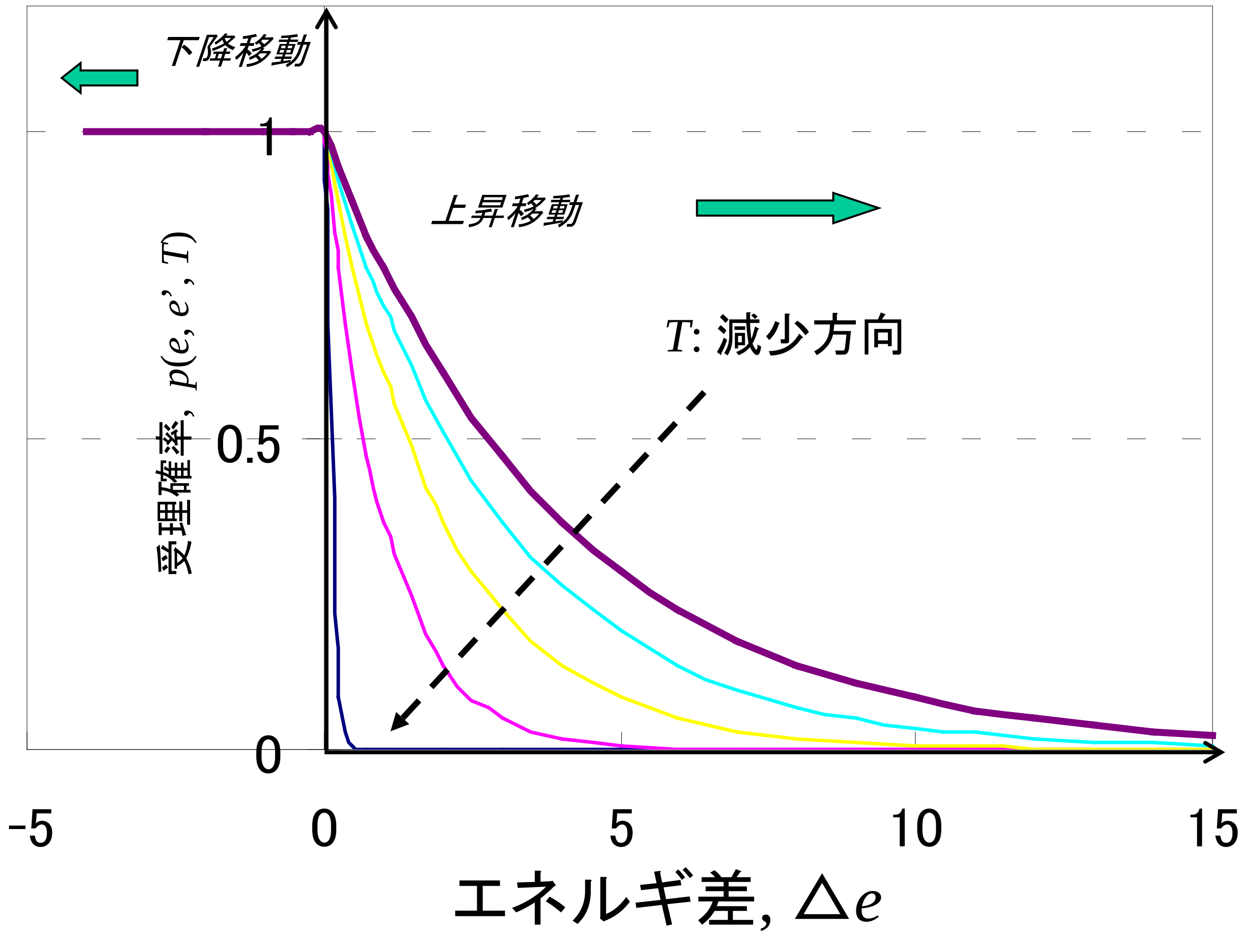
その他の進化手法

擬似アニーリング法（Simulated Annealing; SA）

金属の焼きなましとして知られている物理現象を模擬

焼きなましは，加熱と徐冷によって金属の結晶サイズを大きくて
構造上の欠陥を減らすための良く知られた技術
　加熱⇒原子の運動エネルギは活性化するため，
　　　　エネルギが局所的最小状態から高エネルギの状態に遷移する。
　徐冷⇒最初の状態より低エネルギの安定した位置に移りやすくなる。

SAでは、探索空間の各々の探索点＝物理システムの状態
　　　　最小化される目的関数＝システムの内部エネルギ状態
　　⇒ エネルギ最小の状態に達成した＝最適解が得られた



遷移確率関数の温度依存性

タブーサーチ（Tabu Search; TS）

タブーリスト：特別な探索履歴（記憶構造）を用いるローカルサーチ手法の一つ。

暫定解$\boldsymbol{x}$からその近傍$N(\boldsymbol{x})$の中の最もよい解$\boldsymbol{x}'$への遷移が繰り返し行なわれる。

⇒このような単純な手続きだけでは，
$\boldsymbol{x}$から$\boldsymbol{x}'$へ，そして$\boldsymbol{x}'$から$\boldsymbol{x}$といった解の循環が起こる
⇒循環を避けるために，認知科学の分野での短期記憶に該当するタブーリストを利用する。
タブーリストに含まれている全ての解への移動は，たとえ解の改善が行われる場合でも一定の間は禁止される。

差分進化法（Differential Evolution; DE）

GAの実数コード化手法の一つ
現実の応用において有力な最適化手法
種々の同属手法:3組表記DE/*x*/*y*/*z*/ により分類

- Target vector ($\boldsymbol{x}_{i,G}$)**の発生**
- Mutant vector ($\boldsymbol{v}_{i,G+1}$)**の生成**

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{i1,G} + \boldsymbol{F}(\boldsymbol{x}_{i3,G} - \boldsymbol{x}_{i2,G})$$

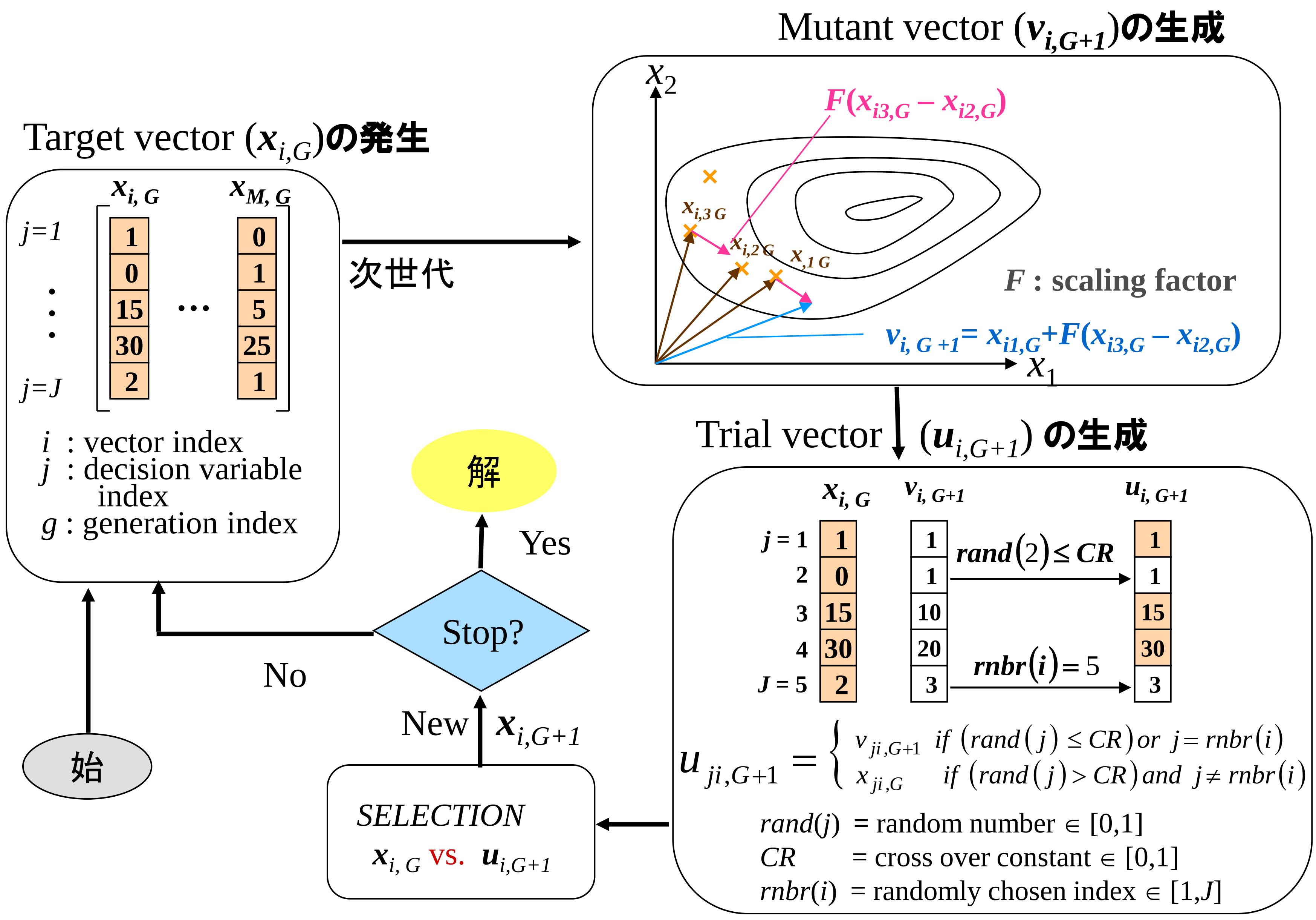
F **: scaling factor**

- Trial vector　($\boldsymbol{u}_{i,G+1}$) **の生成**

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & if\ (rand(j) \le CR)\ or\ \ j = rnbr(i) \\ x_{ji,G} & if\ (rand(j) > CR)\ and\ \ j \ne rnbr(i) \end{cases}$$

$rand(j)$ = random number $\in [0,1]$
CR = cross over constant $\in [0,1]$
$rnbr(i)$ = randomly chosen index $\in [1,J]$

DEの探索イメージ



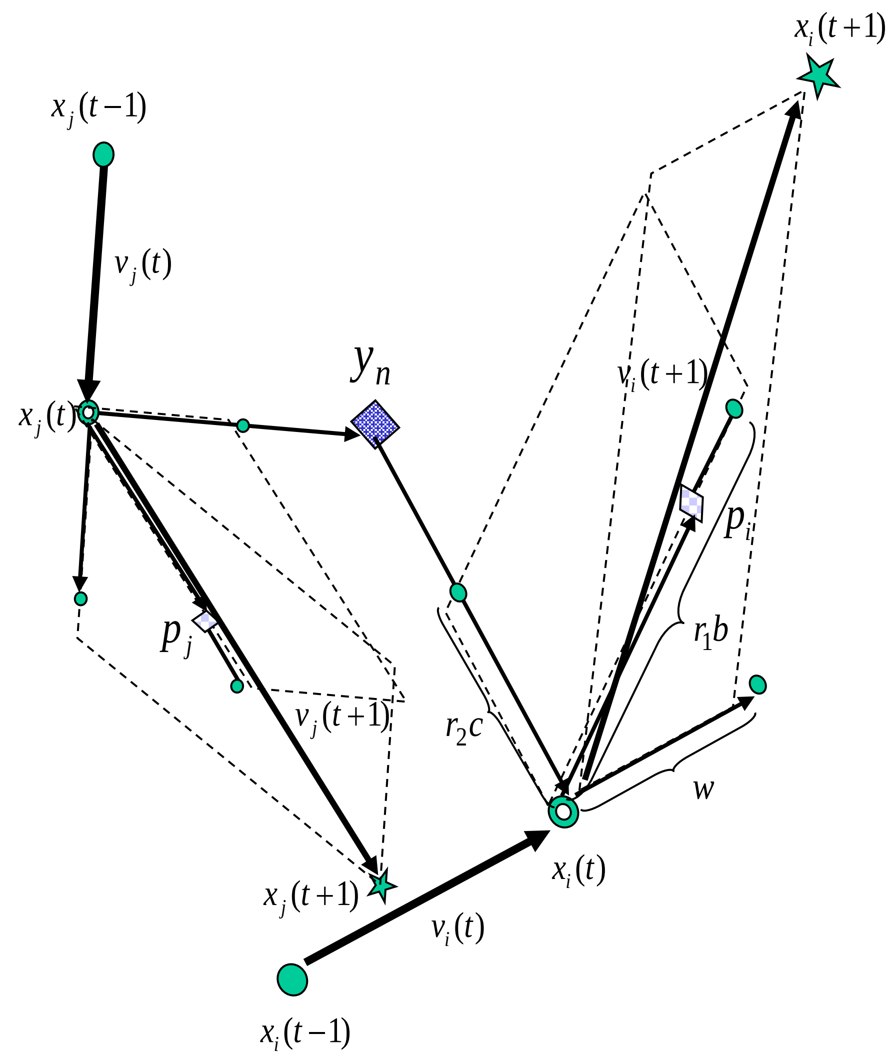
Mutant vector ($v_{i,G+1}$)の生成
x_2
$F(x_{i3,G} - x_{i2,G})$
$x_{i,3\,G}$
$x_{i,2\,G}$
$x_{,1\,G}$
F : scaling factor
$v_{i,G+1} = x_{i1,G} + F(x_{i3,G} - x_{i2,G})$
x_1
Target vector ($x_{i,G}$)の発生
$x_{i,G}$
$x_{M,G}$
$j=1$
$j=J$
1 0 15 30 2
0 1 5 25 1
…
次世代
i : vector index
j : decision variable index
g : generation index
Trial vector ($u_{i,G+1}$) の生成
$x_{i,G}$
$v_{i,G+1}$
$u_{i,G+1}$
$j = 1$ 2 3 4 $J = 5$
1 0 15 30 2
1 1 10 20 3
$rand(2) \le CR$
$rnbr(i) = 5$
1 1 15 30 3
$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (rand(j) \le CR) \text{ or } j = rnbr(i) \\ x_{ji,G} & \text{if } (rand(j) > CR) \text{ and } j \ne rnbr(i) \end{cases}$
$rand(j)$ = random number $\in [0,1]$
CR = cross over constant $\in [0,1]$
$rnbr(i)$ = randomly chosen index $\in [1,J]$
解
Yes
Stop?
No
始
New $x_{i,G+1}$
SELECTION
$x_{i,G}$ vs. $u_{i,G+1}$

粒子群最適化
（Particle Swarm Optimization; PSO）

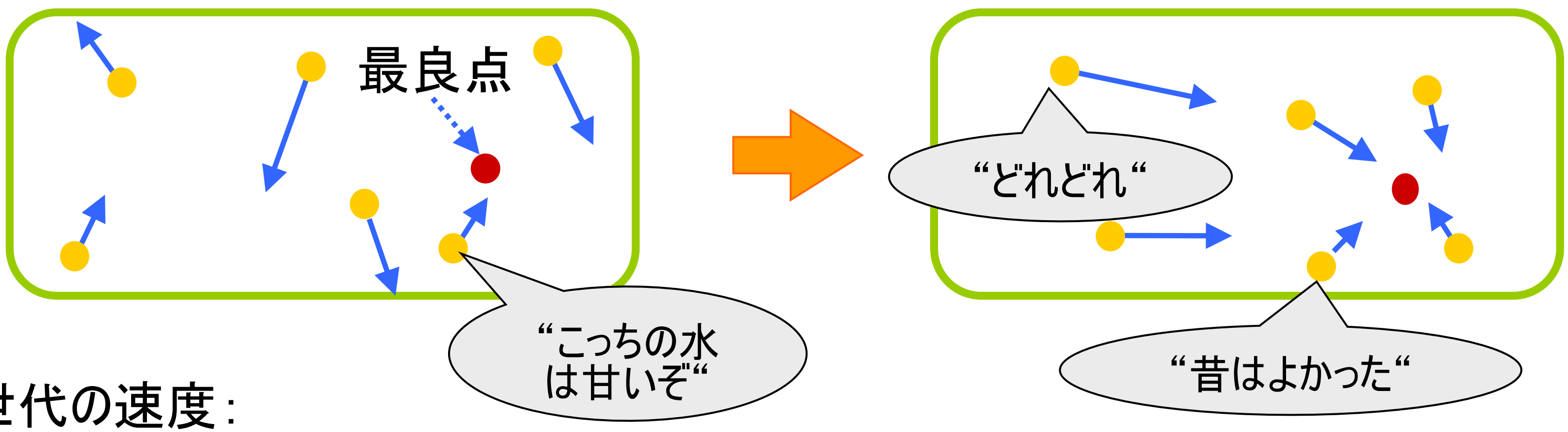
・人工知性研究における自律分散的な集団行動に関わる群の知性の一形態を参考にした，実数コードを用いるメタヒューリスティック手法の一つ。

・原理はC. Reynoldsのボイドに関する理論から派生

・自然界での鳥の群や魚の集団行動を観察していて気付く合目的性にも着目。



PSOの探索イメージ



次世代の速度:

- $$v_i(t+1) = aw \times v_i(t) + b(p_i(t) - x_i(t)) + c(n(t) - x_i(t))$$

現在の速度

自己の過去の最良解との差

集団の最良解との差

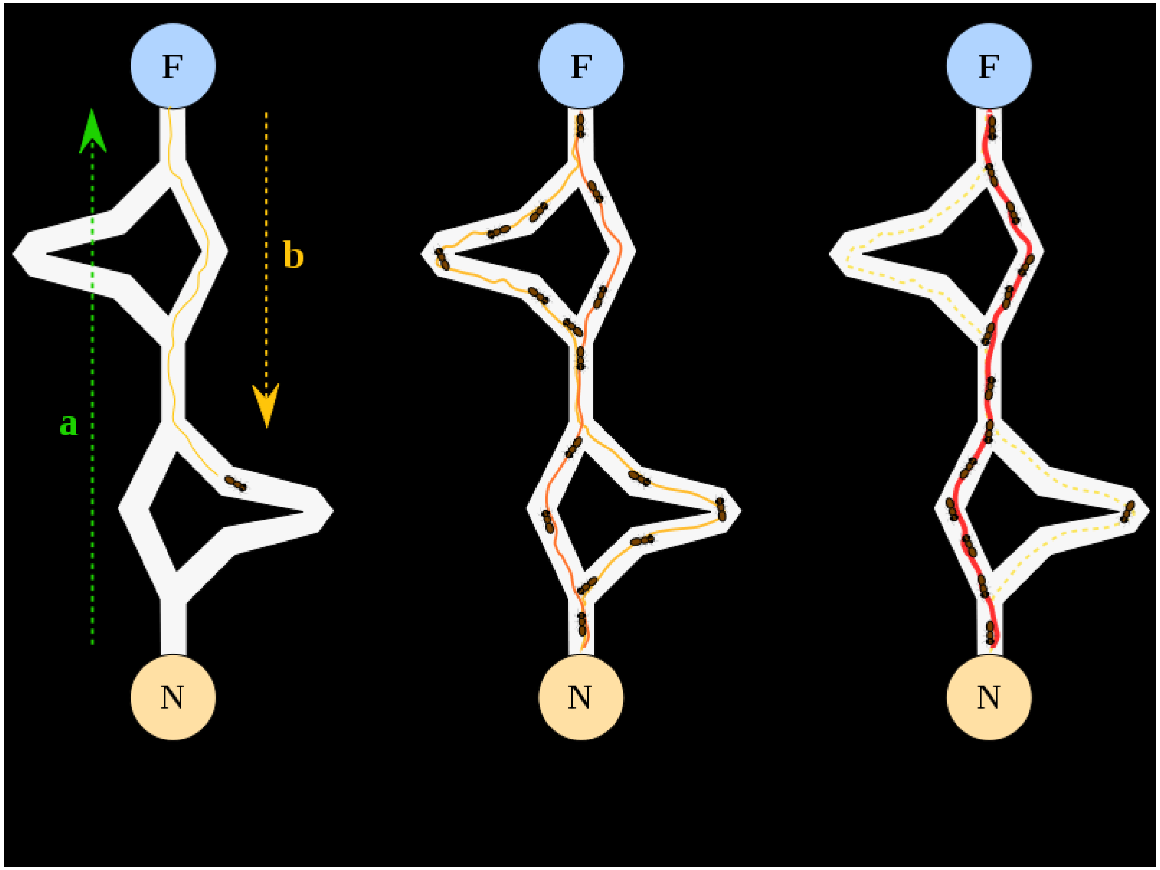
次世代の状態:

- $$x_i(t+1) = x_i(t) + v_i(t+1)$$

t: 世代
M: 個体数
w,a,b,c: 係数
$p_i(t)$: 現在までのi個体の最良位置
$n(t)$: 現在までの集団での最良位置

• ACO (Ant Colony Optimization)

巣から餌への経路を見い出す蟻の行動を模擬した探索法
フェロモンの蒸発で探索を制御

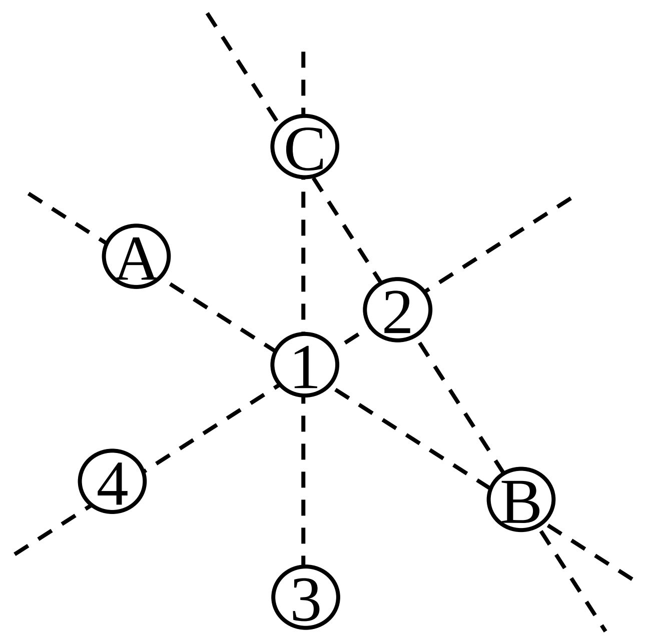


• Memetic Algorithm

GAから派生した方法
交叉の操作とローカルサーチを組み合わせる
遺伝的ローカルサーチとかハイブリッドGAとかよばれる。

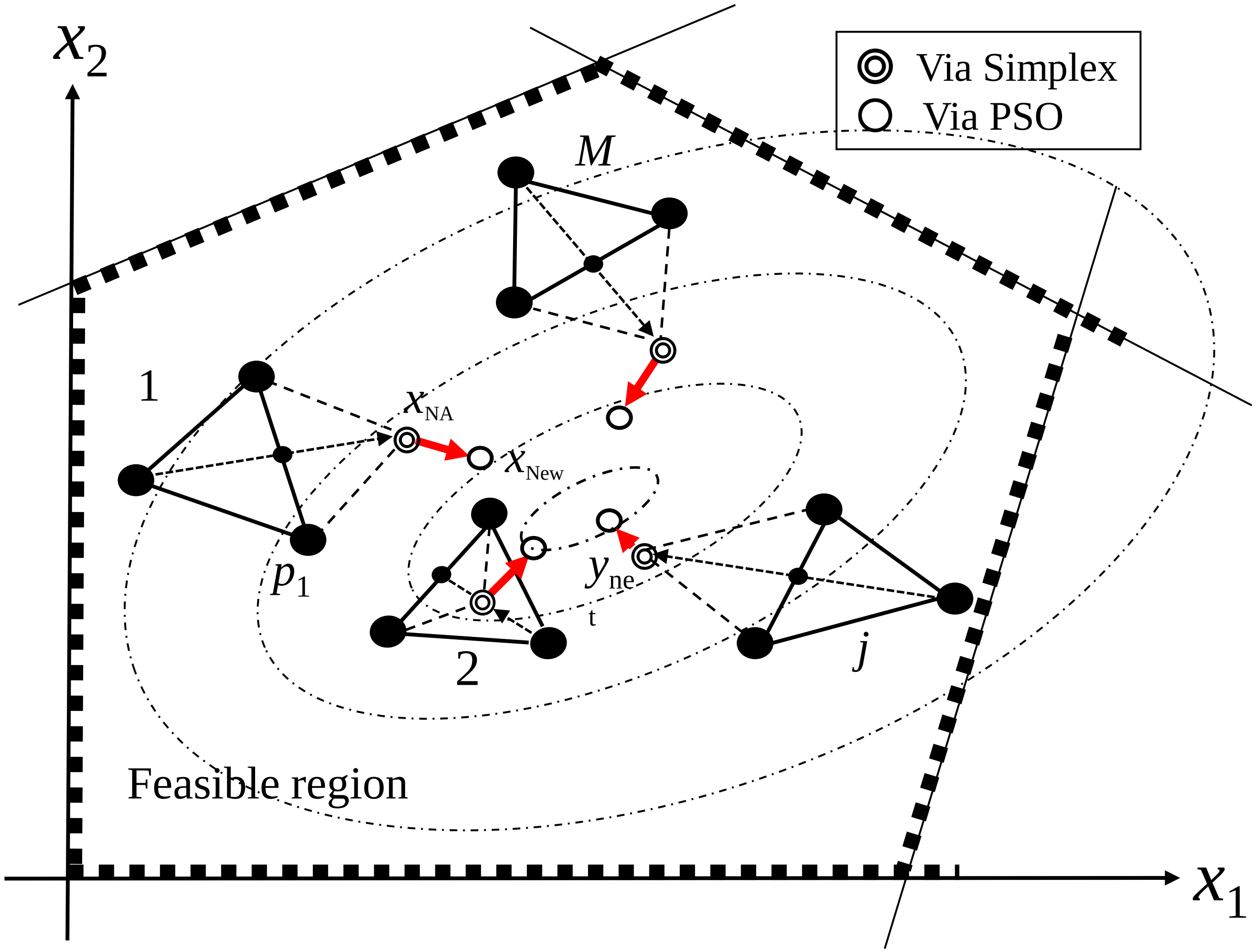
• Scatter Search

一般経路生成（generalized path construction）とよばれる方法に基づいて，参照集合とよばれる解集合を形成し，これを基準として新しい解を生成
参照集合自身も異なる解の凸および非凸結合の両方によって漸次更新



進化型シンプレックス法

シンプレックス法とPSOのハイブリッド化法



補　足

最適化モデル(手法)の役割

ものづくりにおける
競争優位性

イノベーション

高いリスク

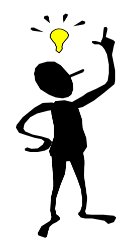
効率的リスク管理

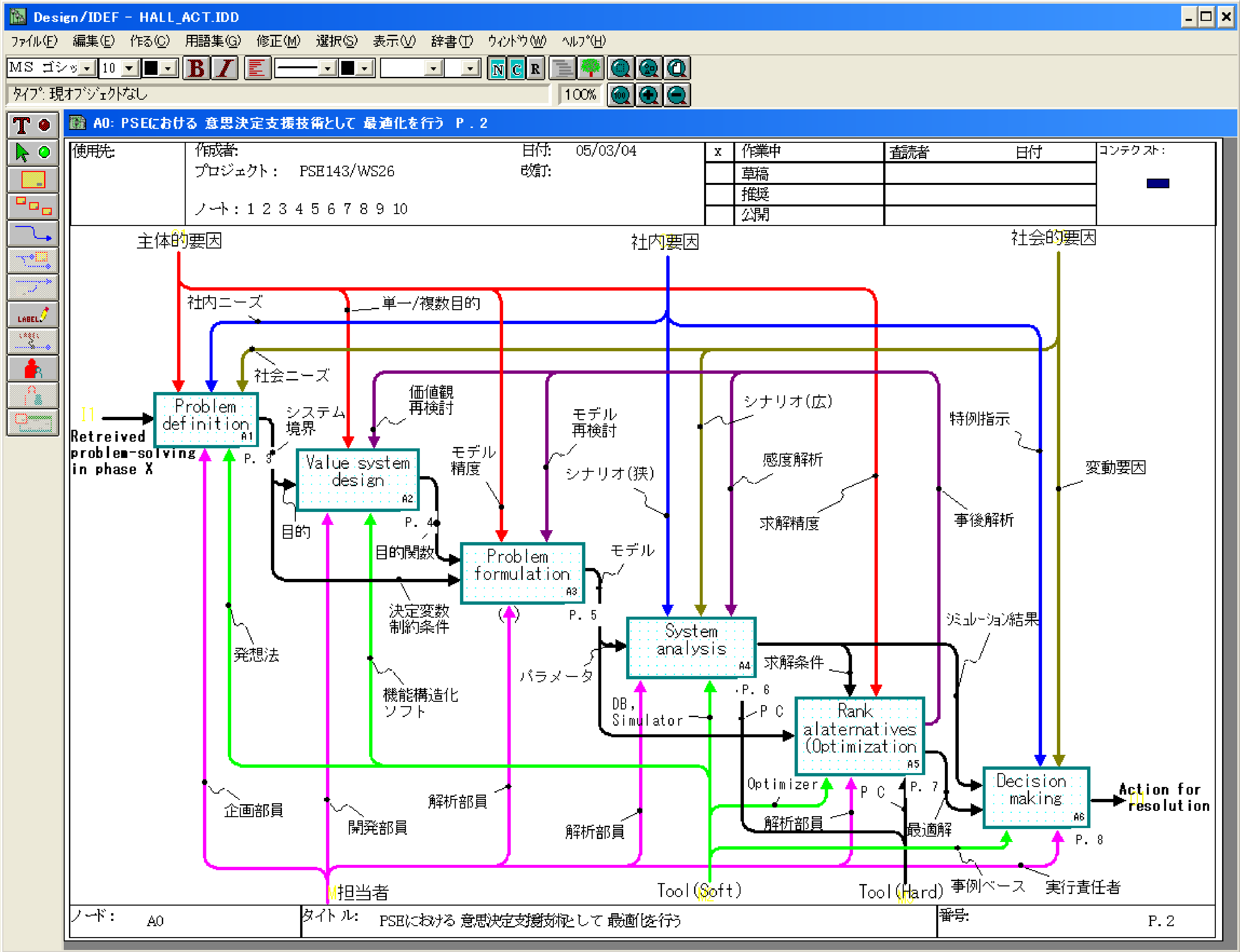
製品リードタイム
の短縮

代替案評価に要する
努力の低減

最適化モデル

合理的意思決定支援





Design/IDEF - HALL_ACT.IDD
ファイル(F) 編集(E) 作る(C) 用語集(G) 修正(M) 選択(S) 表示(V) 辞書(T) ウィンドウ(W) ヘルプ(H)
タイプ: 現オブジェクトなし
100%
A0: PSEにおける 意思決定支援技術として 最適化を行う P . 2
使用先:
作成者:
プロジェクト: PSE143/WS26
ノート: 1 2 3 4 5 6 7 8 9 10
日付: 05/03/04
改訂:
x
作業中
草稿
推奨
公開
査読者
日付
コンテクスト:
主体的要因
社内要因
社会的要因
社内ニーズ
単一/複数目的
社会ニーズ
価値観再検討
モデル再検討
シナリオ(広)
特例指示
I1
Retreived problem-solving in phase X
Problem definition
A1
P. 3
システム境界
Value system design
A2
P. 4
目的
モデル精度
シナリオ(狭)
感度解析
求解精度
事後解析
変動要因
目的関数
Problem formulation
A3
P. 5
モデル
決定変数 制約条件
System analysis
A4
P. 6
シミュレーション結果
求解条件
発想法
機能構造化ソフト
パラメータ
DB , Simulator
P C
Rank alaternatives (Optimization
A5
P. 7
Decision making
A6
P. 8
Action for resolution
O1
企画部員
開発部員
解析部員
解析部員
Optimizer
P C
解析部員
最適解
事例ベース
実行責任者
M1 担当者
Tool(Soft)
M2
Tool(Hard)
M3
ノード: A0
タイトル: PSEにおける 意思決定支援技術として 最適化を行う
番号: P. 2

はじめに

一般的な問題解決手順の例