

実数値遺伝的アルゴリズム

Real-Coded Genetic Algorithm

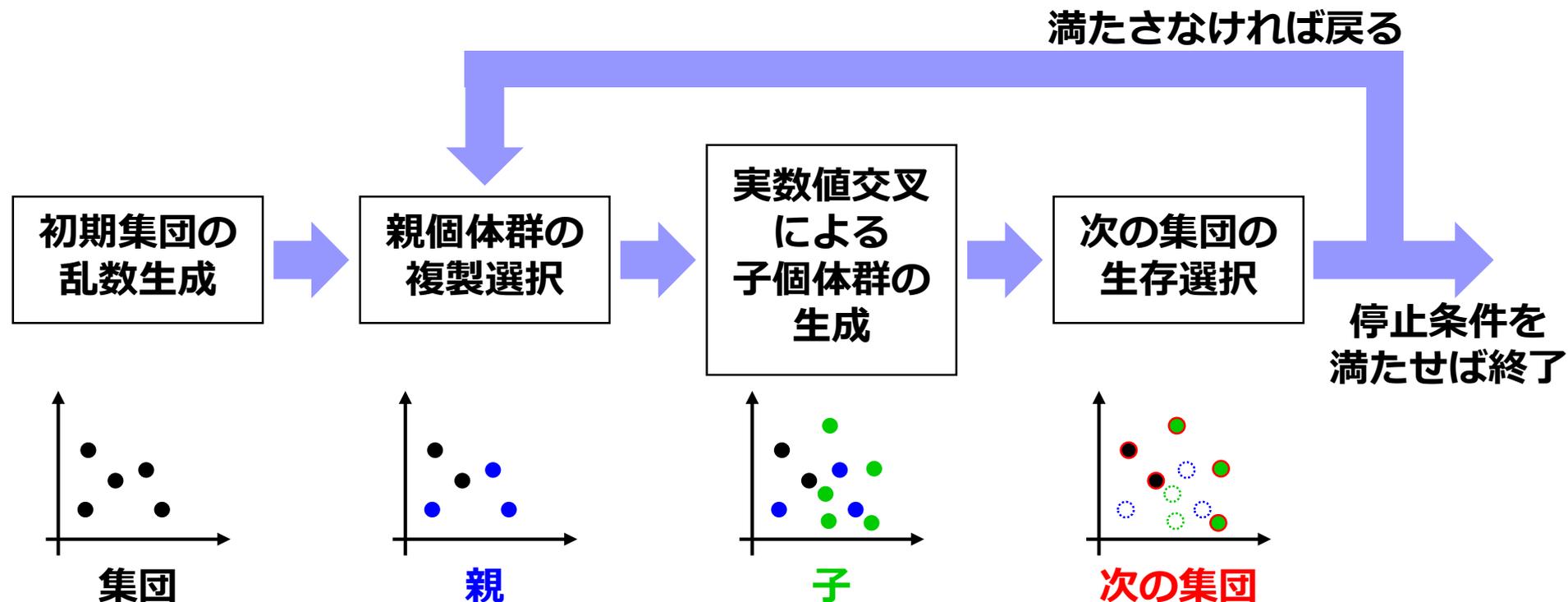
大阪府立大学 工学研究科
清水 悠生

はじめに

- ✓ 実数値遺伝的アルゴリズムでよく使われる手法の紹介をします（単目的最適化がメイン）
- ✓ 数学的な話はあまりしません（文献見て）
- ✓ 本記事で紹介するアルゴリズムは全てGitHubで実装しています（python）
- ✓ <https://github.com/yshimizu12/RealCodedGeneticAlgorithm>

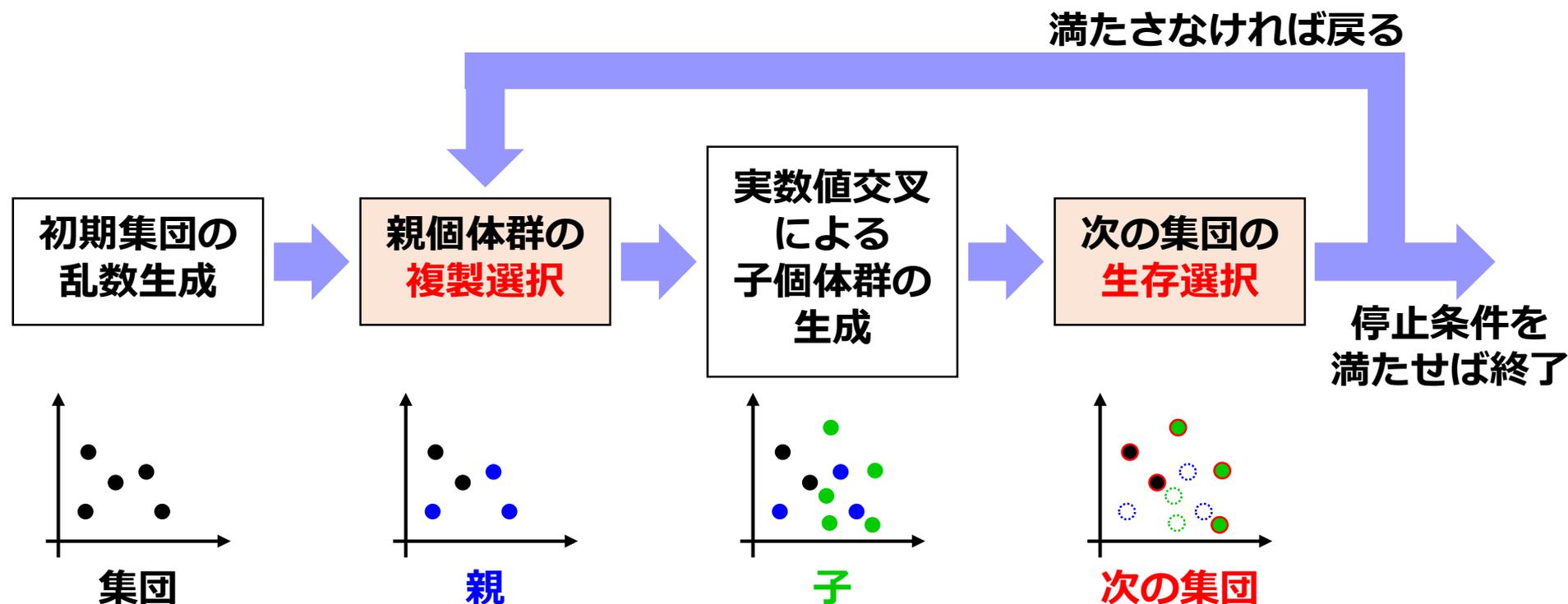
実数値遺伝的アルゴリズムの概要

- ✓ 実数値ベクトルを遺伝子型とする進化計算法を**実数値遺伝的アルゴリズム**(GA: Genetic Algorithm)と呼ぶ
- ✓ 通常のGAとは異なり**実数値ベクトル**を操作する
- ✓ 実数値GAの枠組みは下記のとおり



世代交代モデル

- ✓ 実数値GAの枠組みの中の複製選択と生存選択を合わせて**世代交代モデル**と呼ぶ
- ✓ 本記事では2種類の世代交代モデル (MGG, JGG) を紹介



MGG (Minimal Generation Gap)

- ✓ 世代交代モデルの一つ
- ✓ **親個体数が2個の場合を前提**に設計 (BLX-aなど)
- ✓ 多親を用いた交叉に対しては収束が遅れる恐れあり

複製選択

- ① 集団から親個体をランダムに非復元抽出
(個体数は交叉に依存)

子個体の生成

- ② 親個体群に交叉を繰り返し適用し, 子個体を生成

生存選択

- ③ 親個体群から2個体をランダムに選択し子個体に追加
家族と呼ぶ
- ④ 家族の中から評価値が最良の1個体とルーレット選択により
1個体を選択
- ⑤ 家族に選ばれていない親個体群に選択した2個体を追加
- ⑥ 親個体群を集団に戻し, 次集団とする

JGG (Just Generation Gap)

- ✓ MGGを**多親交叉向けに改良**した世代交代モデル
- ✓ 生存選択の対象を子個体に限定し，親個体は全て淘汰

複製選択

- ① 集団から親個体をランダムに非復元抽出（個体数は交叉に依存）

子個体の生成

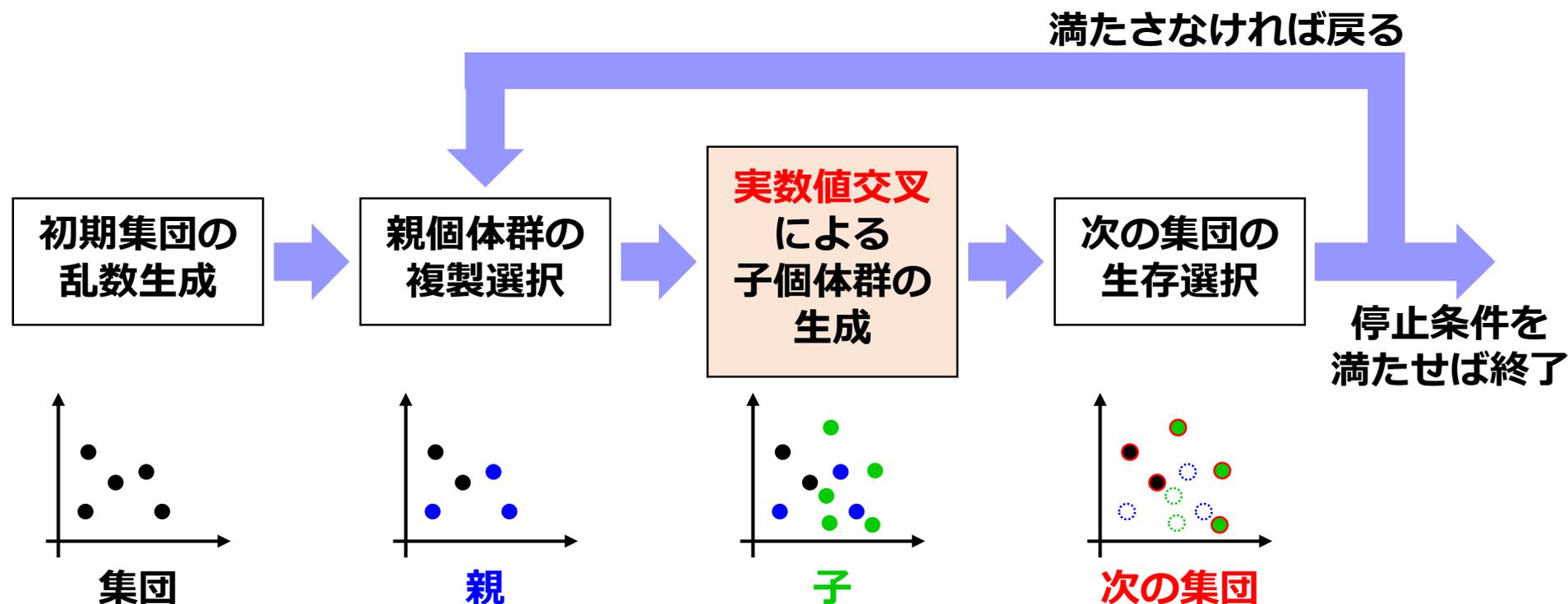
- ② 親個体群に交叉を繰り返し適用し子個体を生成

生存選択

- ③ 子個体群か評価値が上位の個体を親個体の数だけ選択
- ④ 選択した個体を集団に追加次集団とする

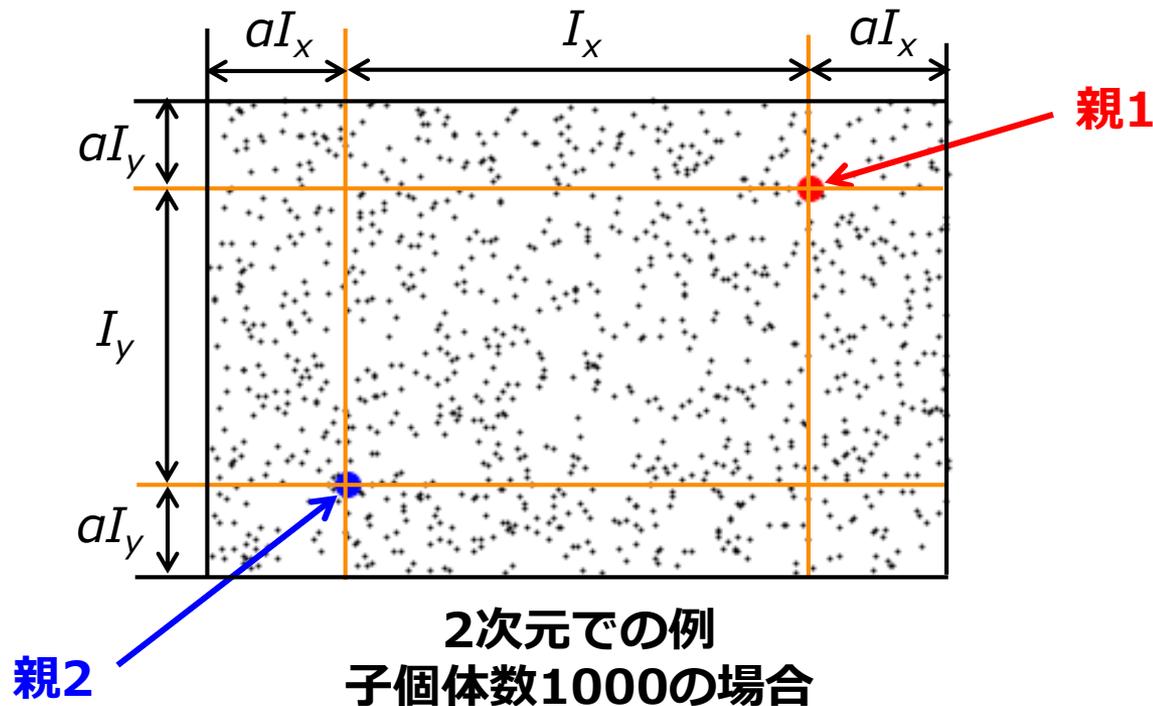
実数値交叉

- ✓ 実数値ベクトルの親個体近傍に子個体を生成する操作を**実数値交叉**と呼ぶ
- ✓ 本記事では5種類の実数値交叉を紹介
(BLX- α , UNDX, UNDX-m, SPX, REXstar)



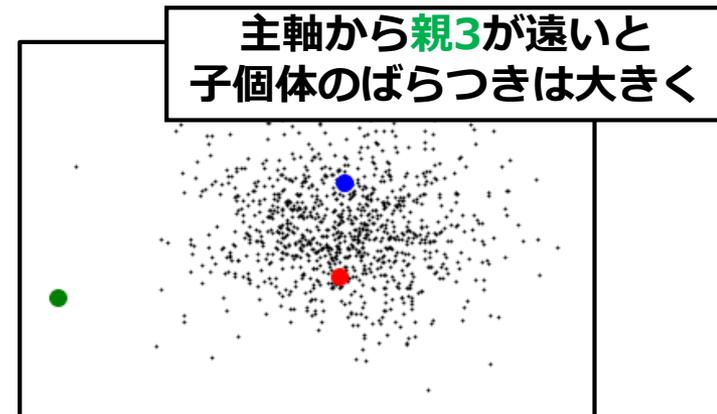
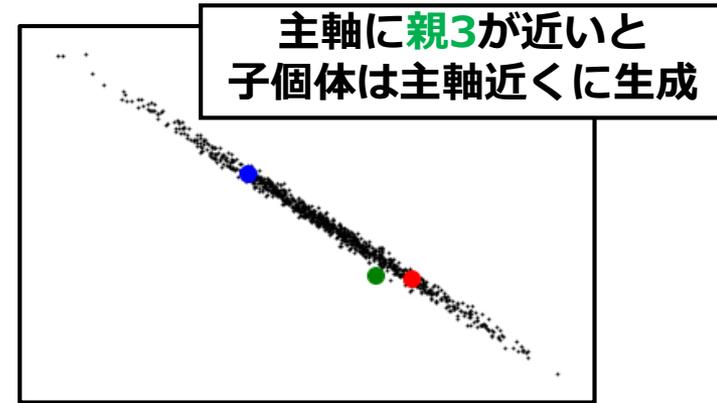
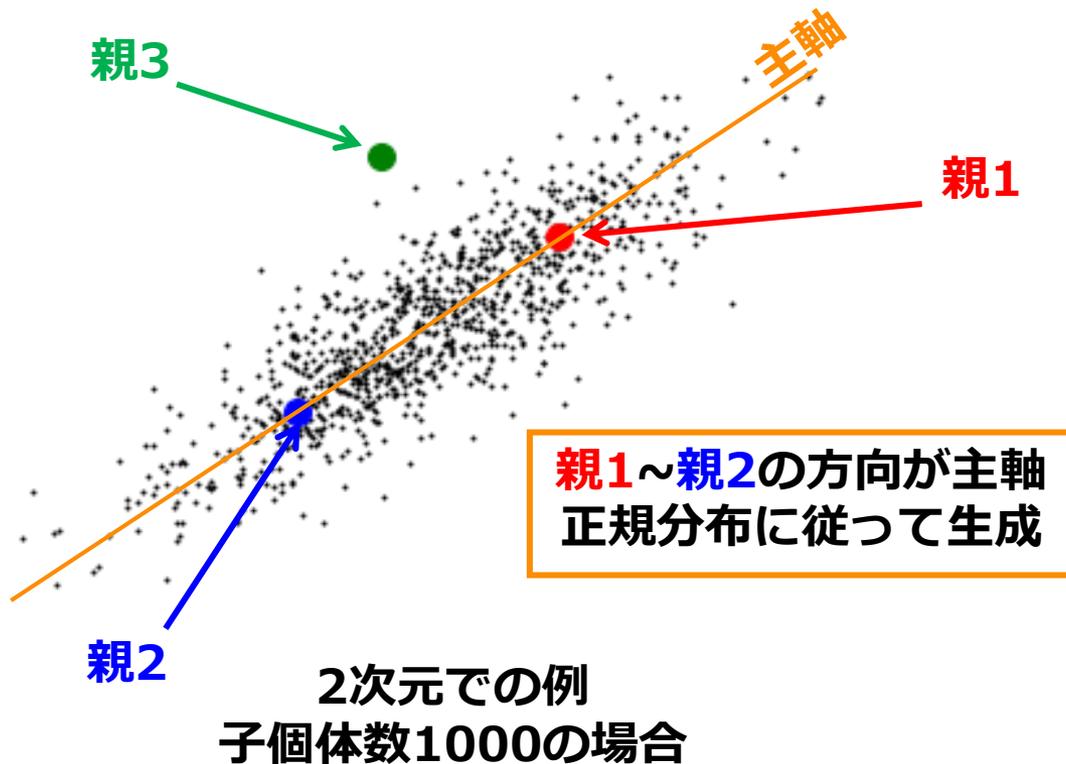
BLX- α (BLeND Crossover alpha)

- ✓ 親個体数は2個
- ✓ 2つの親個体からなる区間の両端を α 倍した領域内にランダムに子個体を生成
- ✓ **悪スケール性の影響は受けにくい**が
変数間依存性の強い関数には対処できない



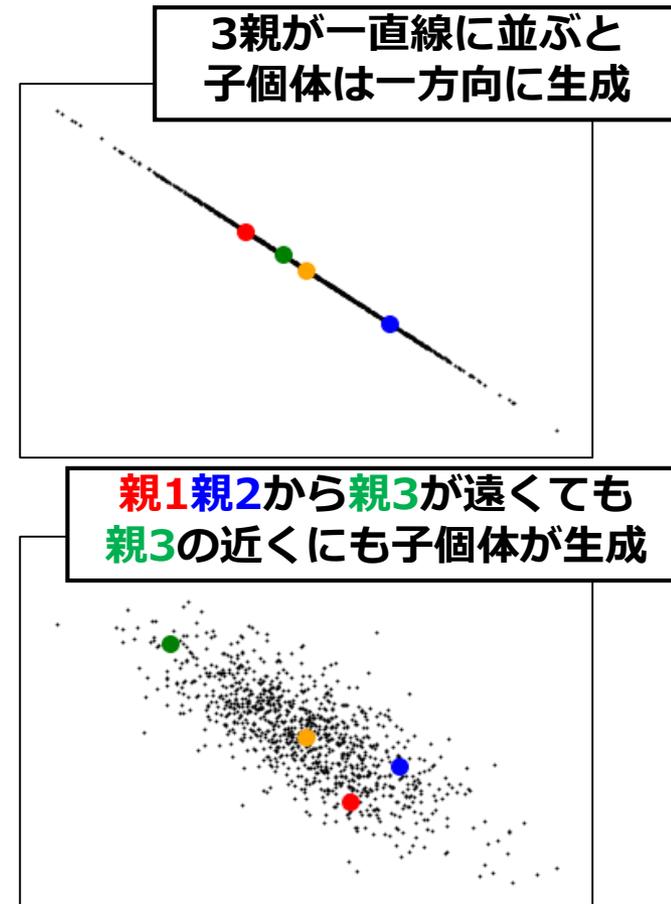
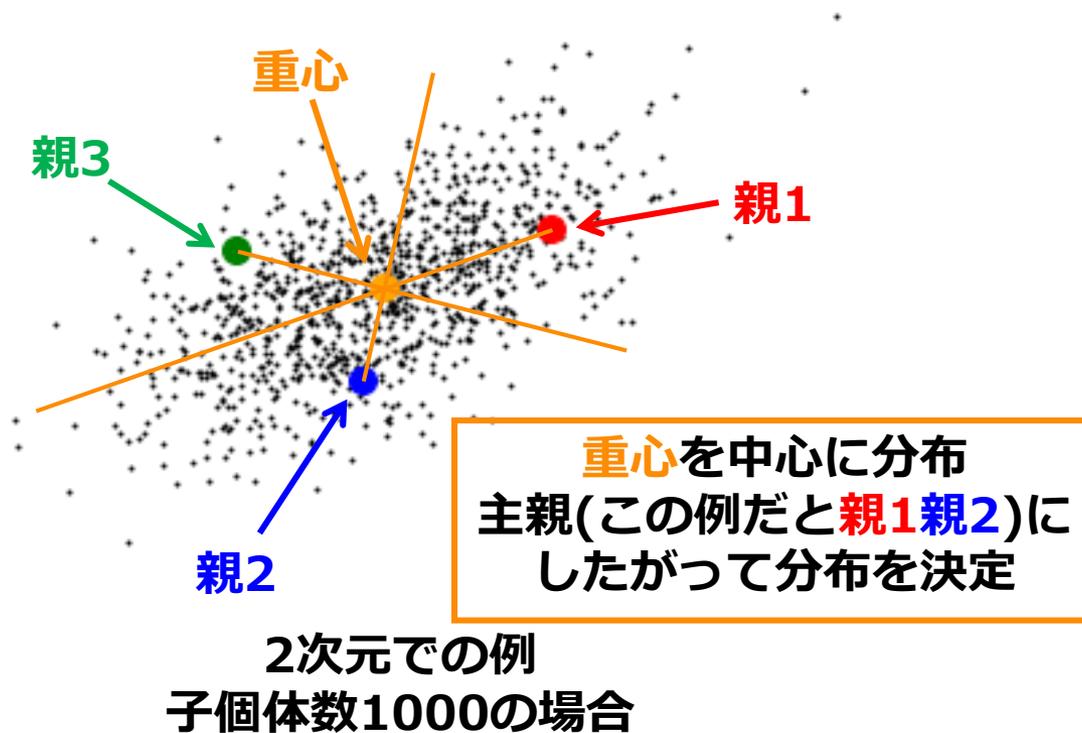
UNDX (Unimodal Normal Distribution Crossover)

- ✓ 親個体数は3個
- ✓ 変数間依存性を考慮した交叉手法
- ✓ 悪スケール性の強い関数に対処できない



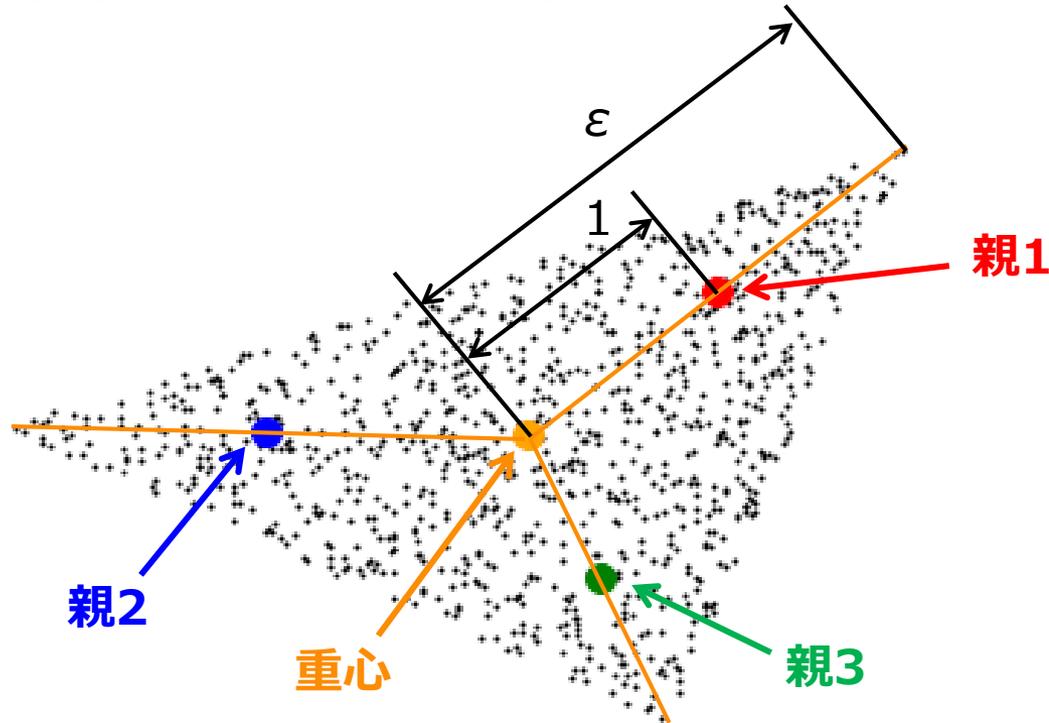
UNDX-m

- ✓ 親個体数は(入力変数の次元+1)個
- ✓ UNDXのスケール依存性の改善を意図した多親交叉
- ✓ **UNDXより変数間依存性や悪スケール性に強い**



SPX (SimPlex Crossover)

- ✓ 親個体数は(入力変数の次元+1)個
- ✓ 重心-親個体ベクトルを ε 倍した点により張られる単体 (simplex)の内部に子個体をランダム生成
- ✓ 変数間依存性や悪スケール性に強い



2次元での例
子個体数1000の場合

REXstar (Real-coded Ensemble Crossover star)

- ✓ 親個体数は(入力変数の次元+1)個
- ✓ SPXの重心に対する非対称性を排除するため
親個体の重心に対する**鏡像点**を設定
- ✓ 親個体と鏡像点の内, 評価値の高い点を中心に子個体を生成
- ✓ **変数間依存性や悪スケール性に強い**

