

適応的個体間距離に基づく 複数解探索型Bat Algorithm

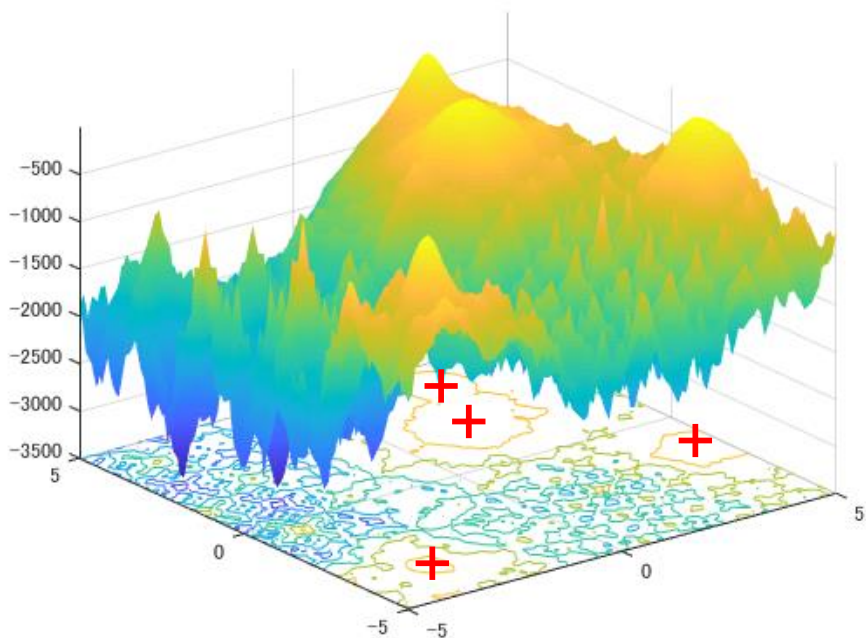
平成30年度 修論発表
情報学専攻 高玉研究室
1730022 岩瀬 拓哉

2019/02/07 (木)

背景: 複数解探索問題

2

多峰性最適化関数



+ : 最適解及び局所解の位置

悪

評価値

良

実問題を
関数化

例) 探査機の月面着陸地点選定問題

入力変数: 位置座標
評価値: 通信可能時間

最適解が着陸困難な
場所だった場合

通信条件が良好な地点
● 最適解 ● 局所解

有力な情報を持つ解を選択肢として複数保持しておくことが重要

多点探索アルゴリズム

3

集団が評価値の高い方へ収束する探索



問題点

一つの最適解あるいは局所解に収束

目的

『多点探索アルゴリズム + 複数局所解保持機構』の提案と有効性の検証

Bat Algorithm (BA) [X. S. Yang, 2010]

4

多点探索アルゴリズムの一つであり，大域探索と局所探索を同時に実施

STEP1

- 最良個体方向へ新たに個体候補を生成

STEP2

- 最良個体周辺に個体候補生成

STEP3

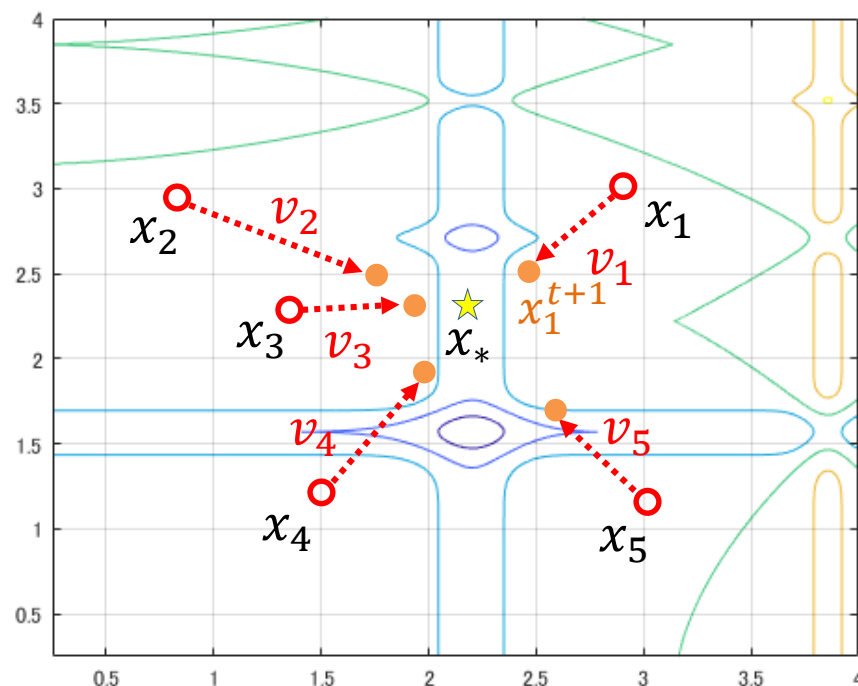
- 探索空間内にランダムで個体候補を生成

STEP4

- 個体候補の評価

STEP5

- 終了条件を満たすまでSTEP1へ戻る



○ : 個体 (コウモリ) x_i ● : 個体候補
★ : 最良個体 x_*

Bat Algorithm (BA) [X. S. Yang, 2010]

4

多点探索アルゴリズムの一つであり，大域探索と局所探索を同時に実施

STEP1

- 最良個体方向へ新たに個体候補を生成

STEP2

- 最良個体周辺に個体候補生成

STEP3

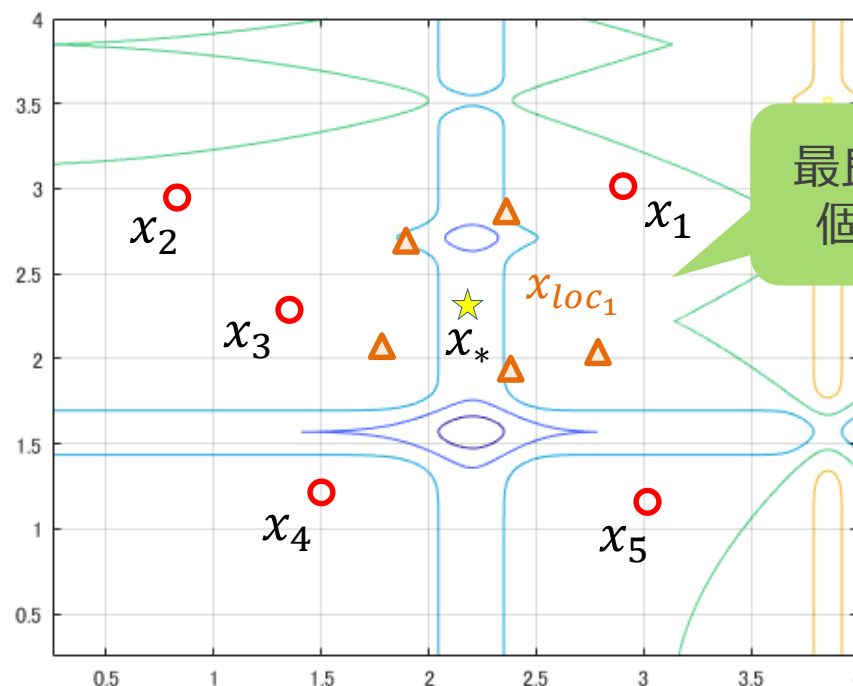
- 探索空間内にランダムで個体候補を生成

STEP4

- 個体候補の評価

STEP5

- 終了条件を満たすまでSTEP1へ戻る



最良個体 x_* 周辺に
個体候補を生成

○ : 個体 (コウモリ) x_i △ : 個体候補
★ : 最良個体 x_*

Bat Algorithm (BA) [X. S. Yang, 2010]

4

多点探索アルゴリズムの一つであり，大域探索と局所探索を同時に実施

STEP1

- 最良個体方向へ新たに個体候補を生成

STEP2

- 最良個体周辺に個体候補生成

STEP3

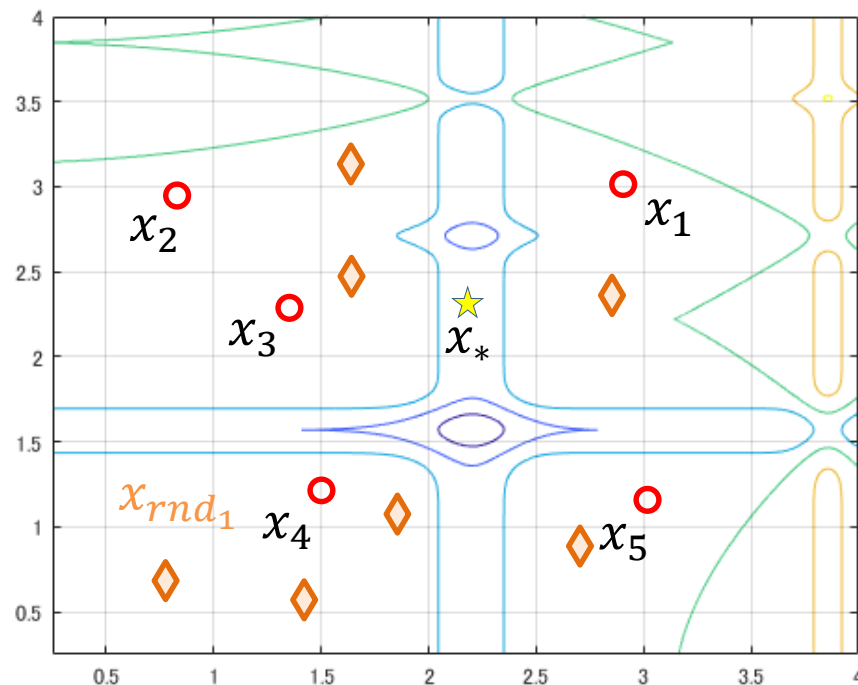
- 探索空間内にランダムで個体候補を生成

STEP4

- 個体候補の評価

STEP5

- 終了条件を満たすまでSTEP1へ戻る



○ : 個体 (コウモリ) x_i ◇ : 個体候補
★ : 最良個体 x_*

Bat Algorithm (BA) [X. S. Yang, 2010]

4

多点探索アルゴリズムの一つであり，大域探索と局所探索を同時に実施

STEP1

- 最良個体方向へ新たに個体候補を生成

STEP2

- 最良個体周辺に個体候補生成

STEP3

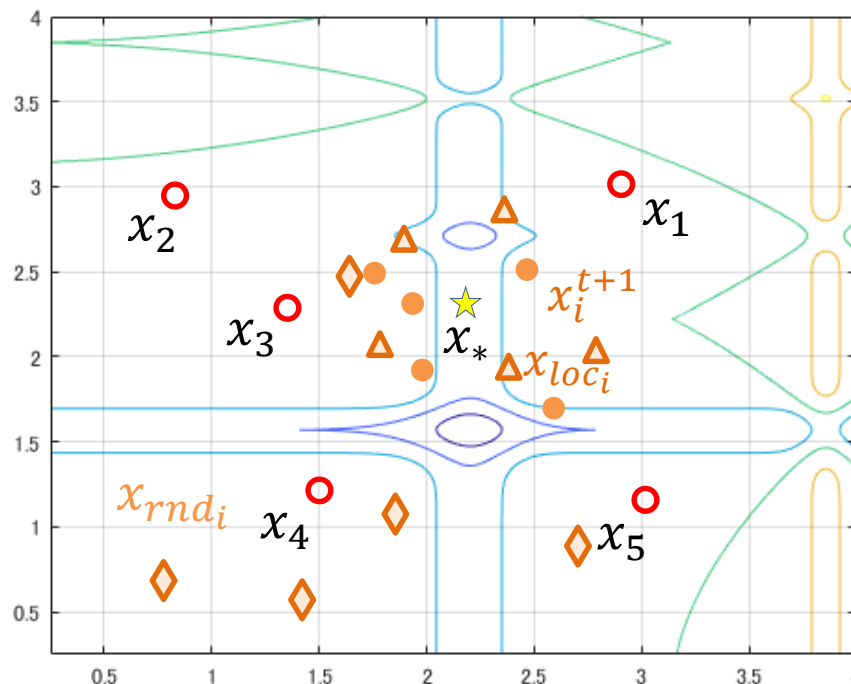
- 探索空間内にランダムで個体候補を生成

STEP4

- 個体候補の評価

STEP5

- 終了条件を満たすまでSTEP1へ戻る



Bat Algorithm (BA) [X. S. Yang, 2010]

4

多点探索アルゴリズムの一つであり，大域探索と局所探索を同時に実施

STEP1

- 最良個体方向へ新たに個体候補を生成

STEP2

- 最良個体周辺に個体候補生成

STEP3

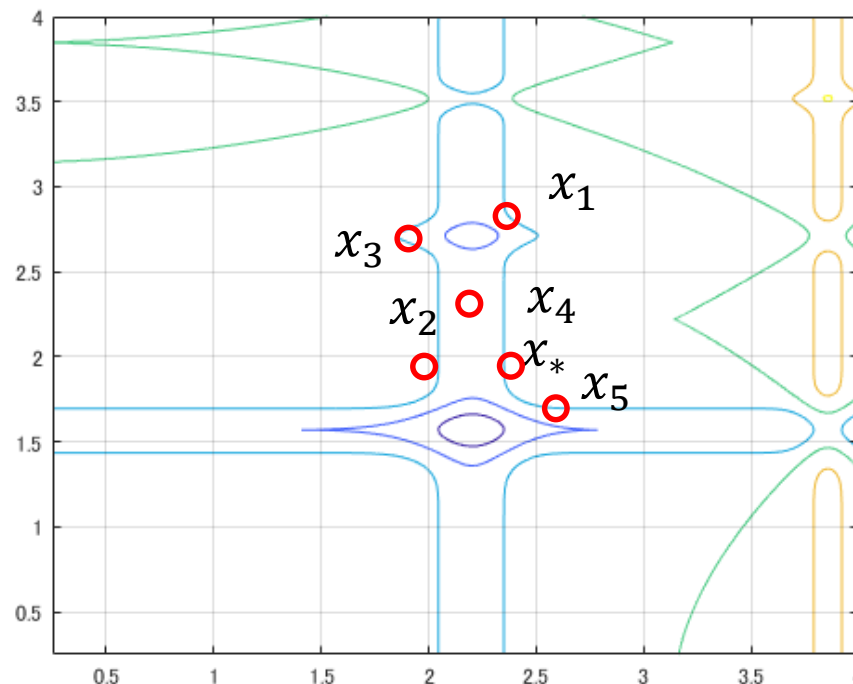
- 探索空間内にランダムで個体候補を生成

STEP4

- 個体候補の評価

STEP5

- 終了条件を満たすまでSTEP1へ戻る



従来の問題に対するアプローチ

5

多点探索アルゴリズム + 複数解探索する機構

Novelty Search-based Bat Algorithm (NSBA)

未探索空間に新しく個体候補を生成する

BA

+

Novelty Search

問題点 保持していた局所解からより良い評価値を持つ解へ移動する

Niche Radius-based Bat Algorithm (NRBA)

探索範囲を分割し、同じ局所解に個体を収束させない機構

BA

+

Niche radius

問題点 評価値の低い局所解で留まる

Dynamic Niche Radius-based Bat Algorithm (DNRBA)

分割した探索範囲内の個体分布密度により探索方法を切り替える

BA

+

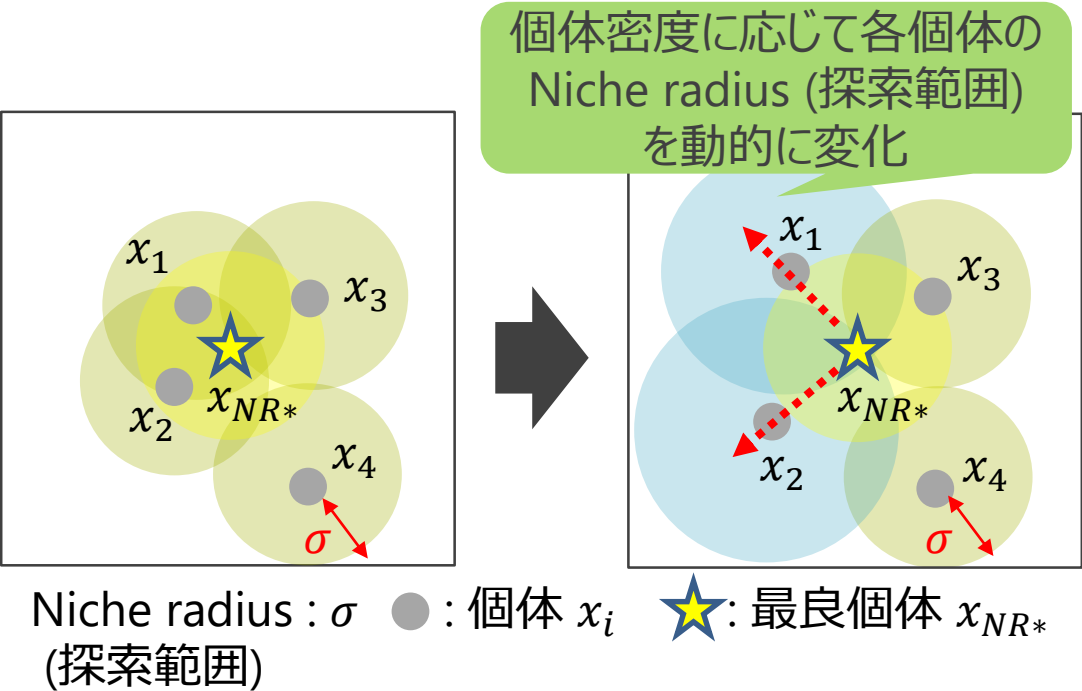
Dynamic Niche
radius

多点探索アルゴリズム + 複数解探索する機構

Niche radius [D.E. Goldberg, et. al, 1987]:
探索空間の大きさと個体数に基づいて算出される

Dynamic Niche Radius-based Bat Algorithm (DNRBA)

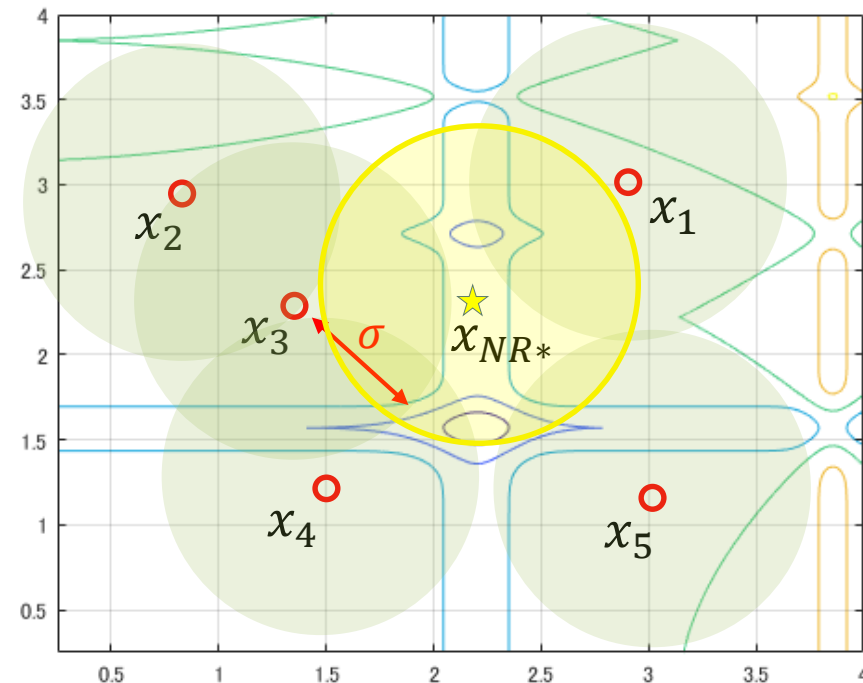
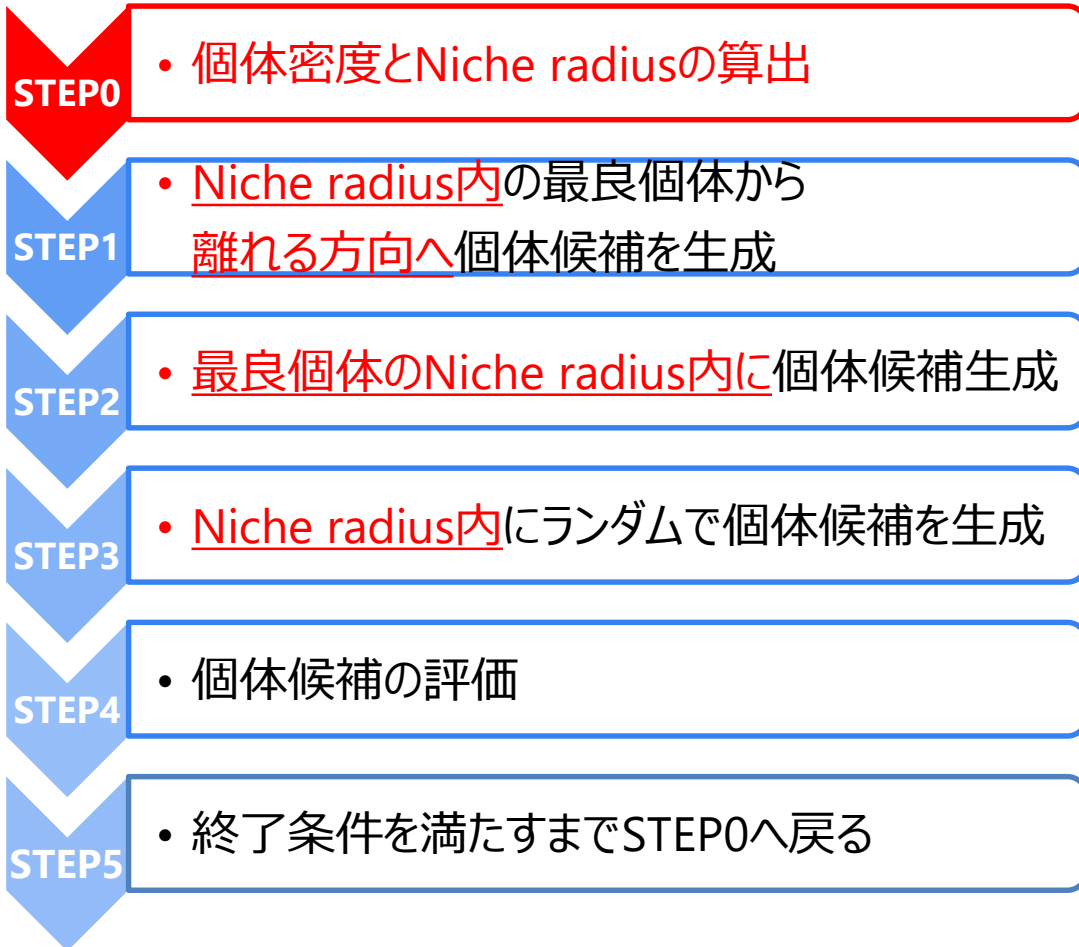
探索範囲内の個体分布密度に応じて動的に探索範囲を調節する



	BA	DNRBA
収束方向への探索	最良個体 方向へ移動	<u>分割した探索範囲</u> (Niche radius)内の最良 個体から <u>離れる方向</u> へ移動
局所探索	最良個体 付近	<u>Niche radius内</u> の最良個体
ランダム探索	解探索空間	<u>Niche radius内</u>
アルゴリズム の特徴	全個体が 最適解へ 収束	探索空間内の局所解 も捕捉可能

DNRBAのフローチャート

7



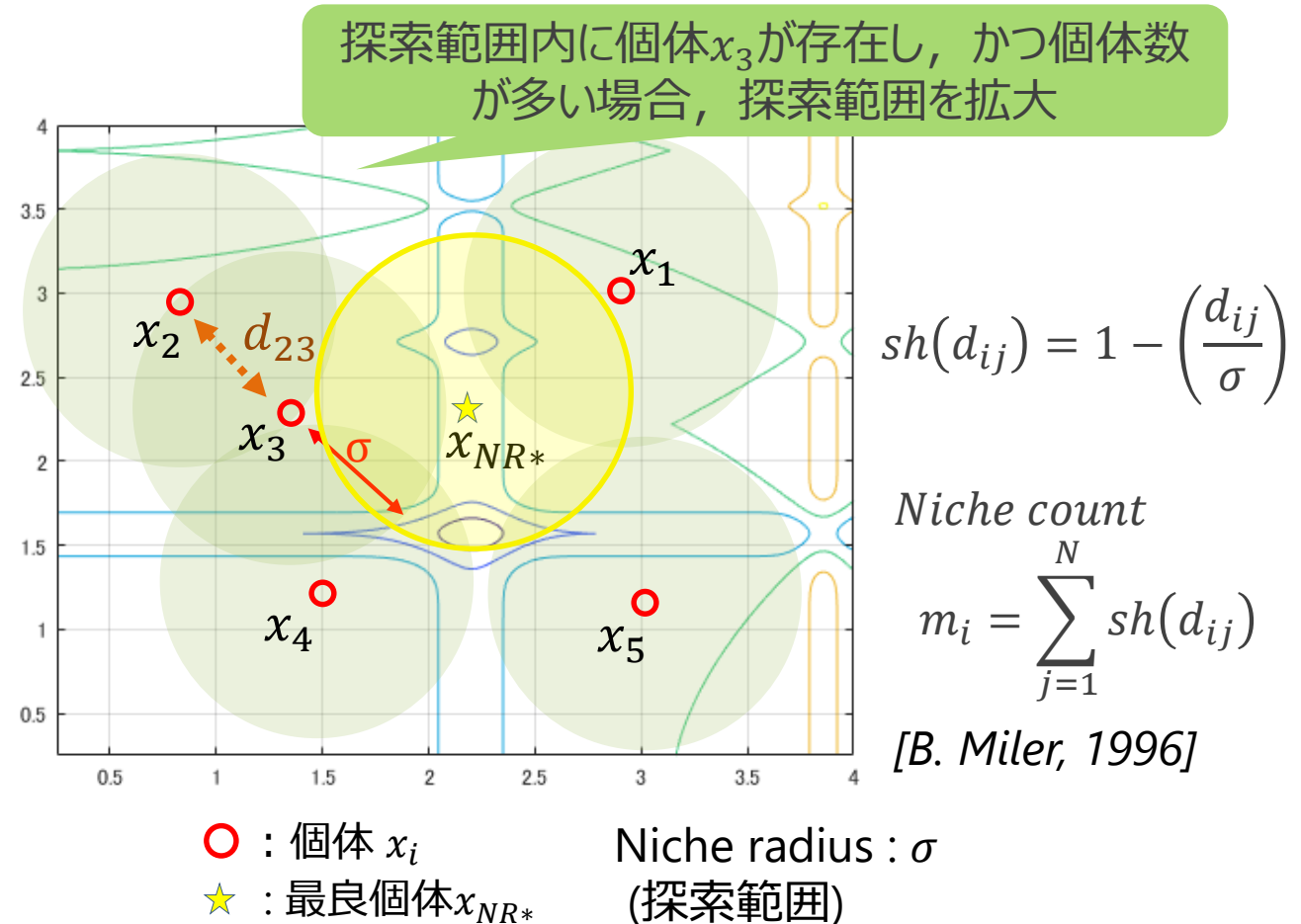
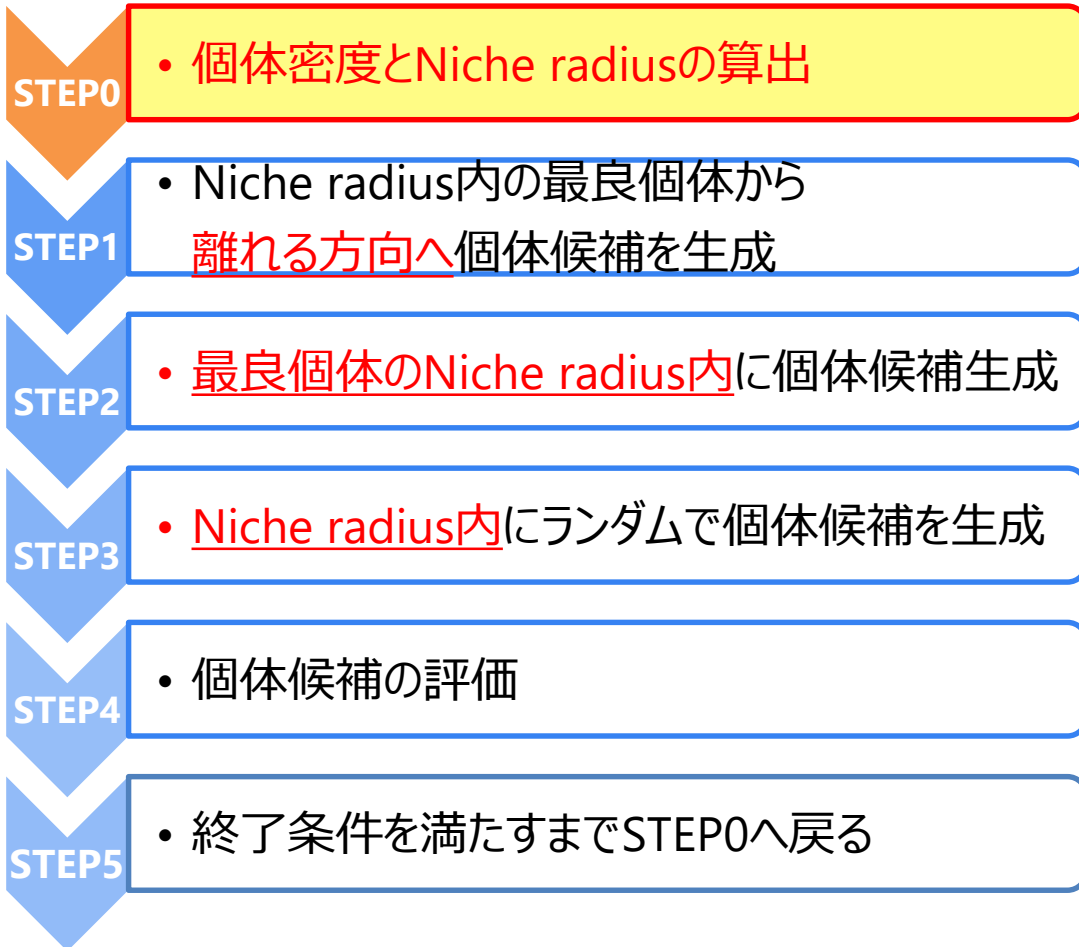
○ : 個体 x_i

★ : 最良個体 x_{NR*}

Niche radius : σ
(探索範囲)

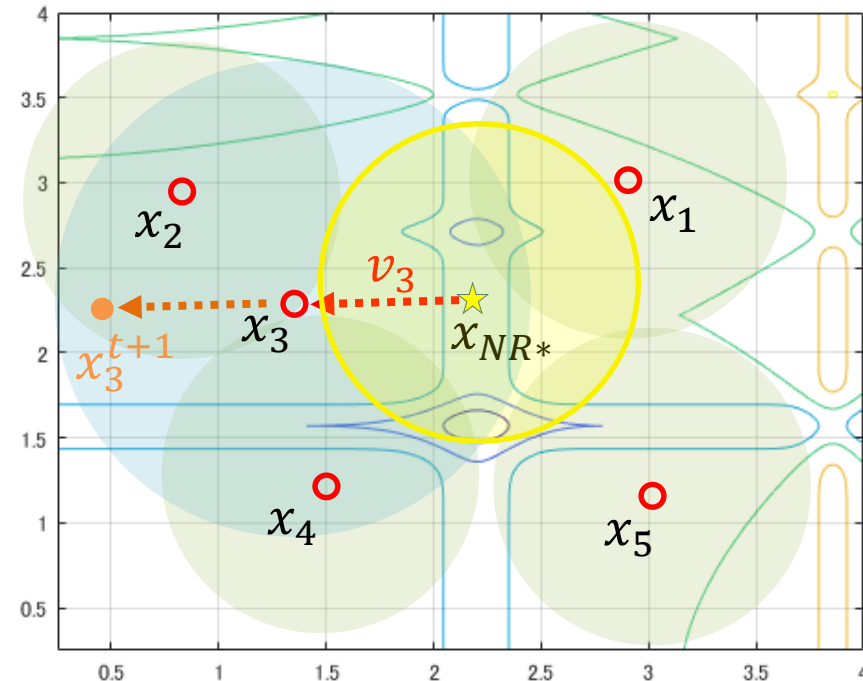
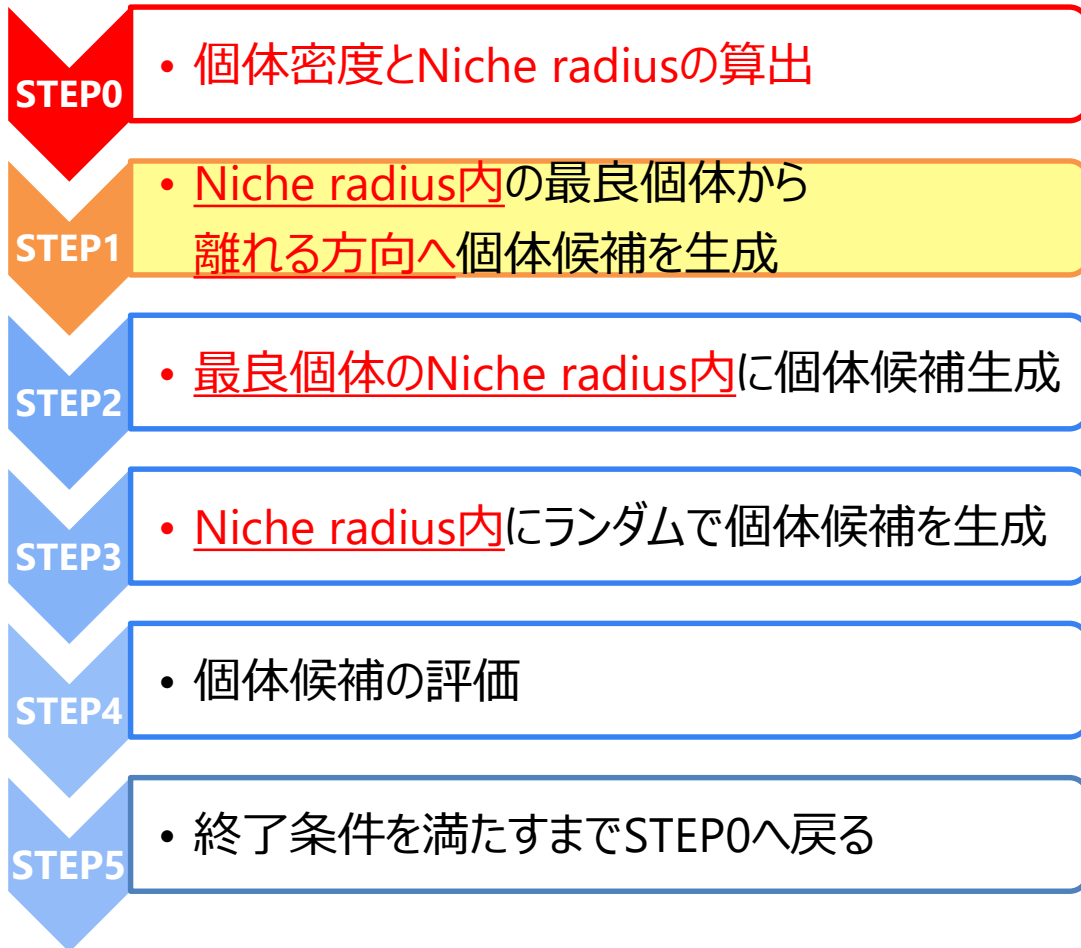
DNRBAのフローチャート

7



DNRBAのフローチャート

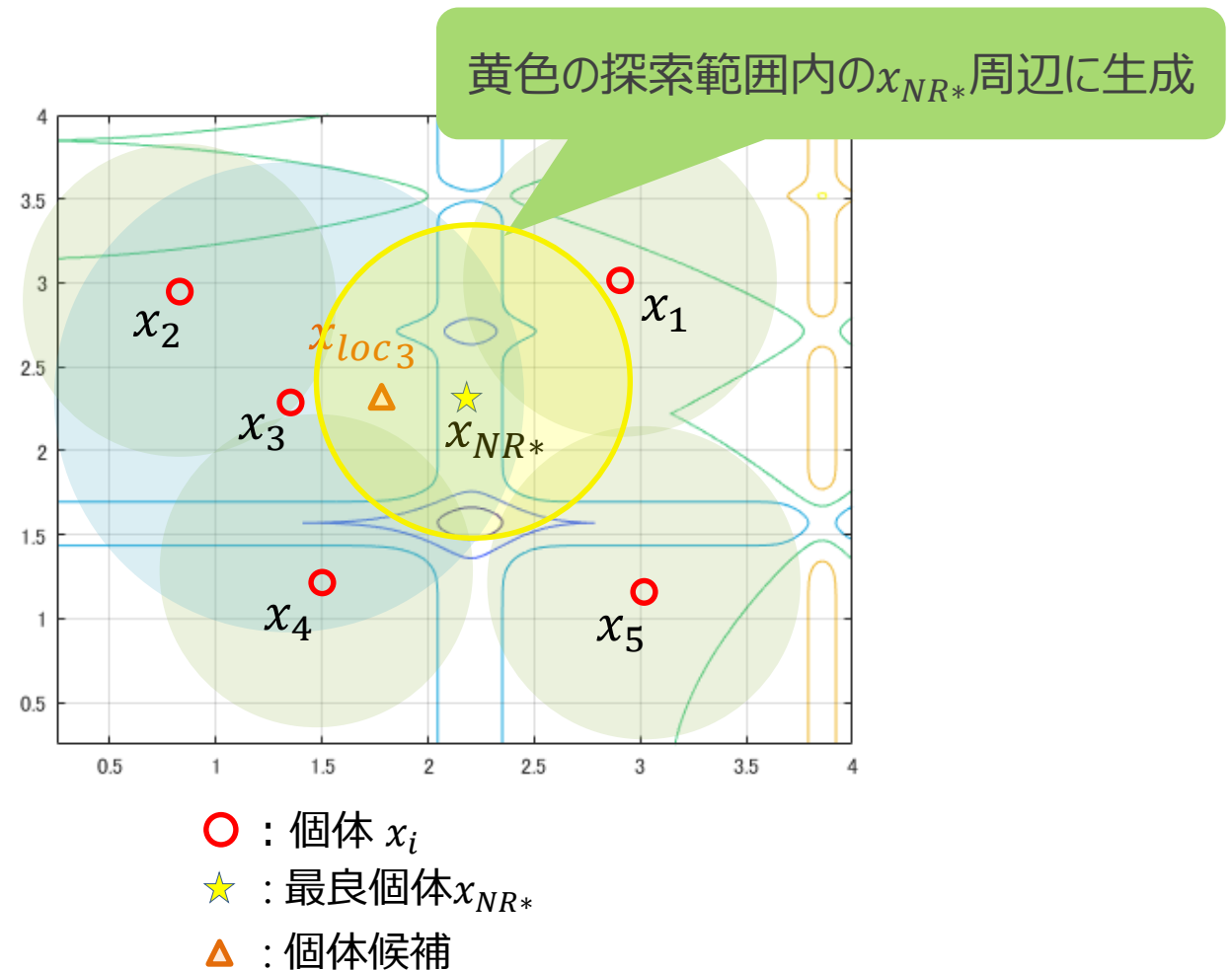
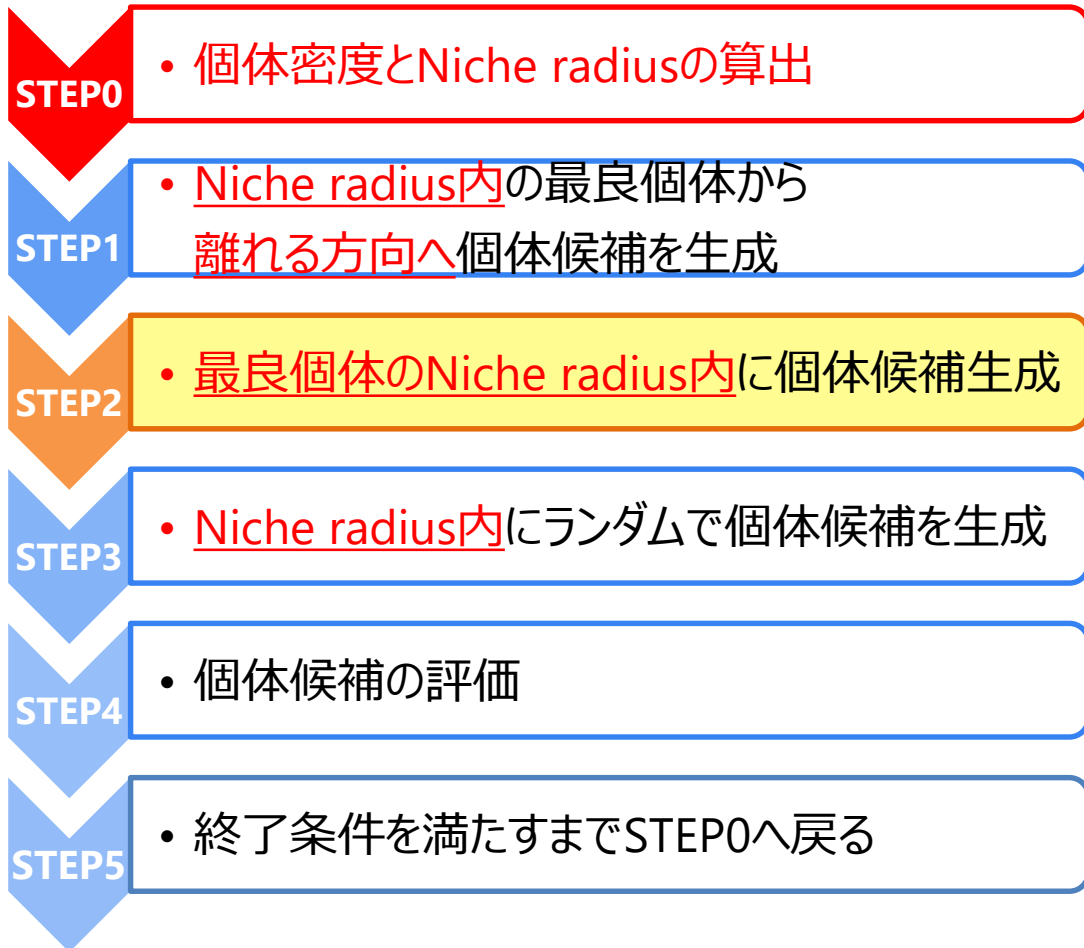
7



- : 個体 x_i
- ★ : 最良個体 x_{NR*}
- : 個体候補

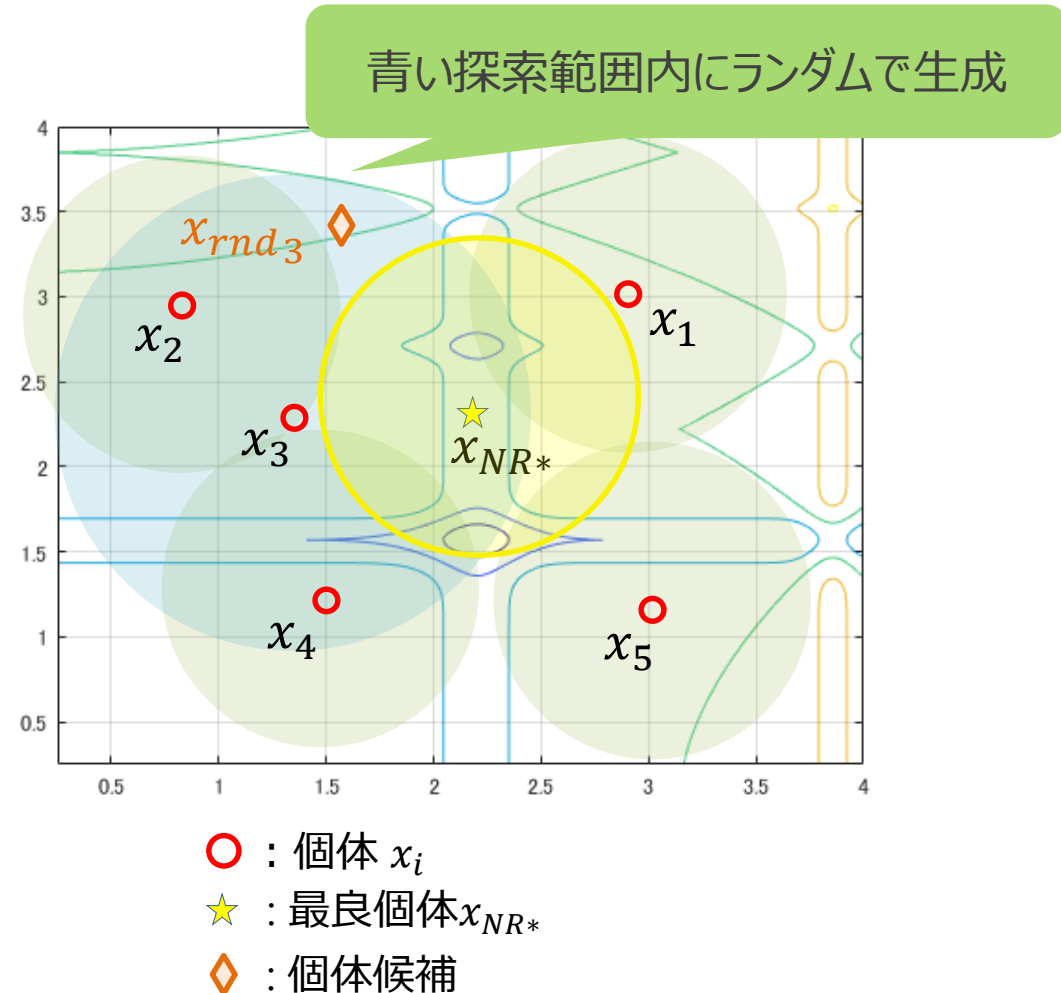
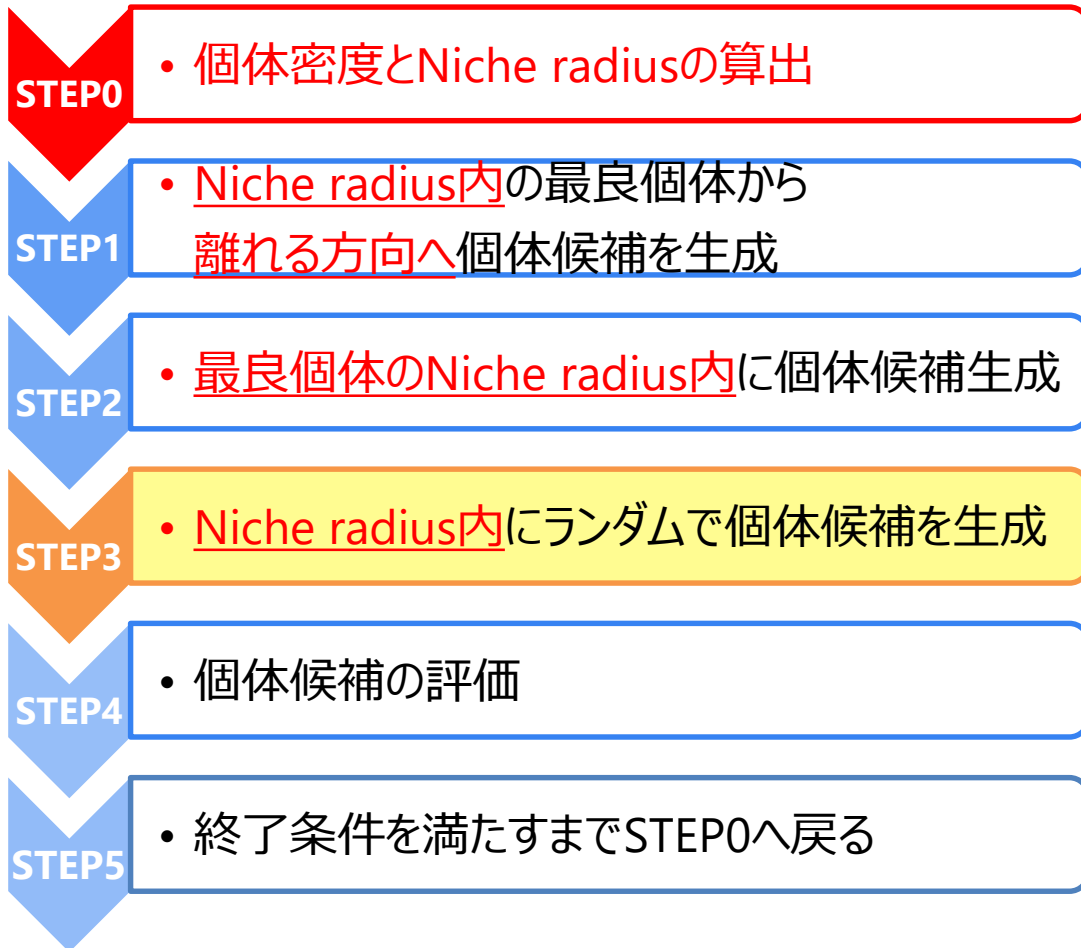
DNRBAのフローチャート

7



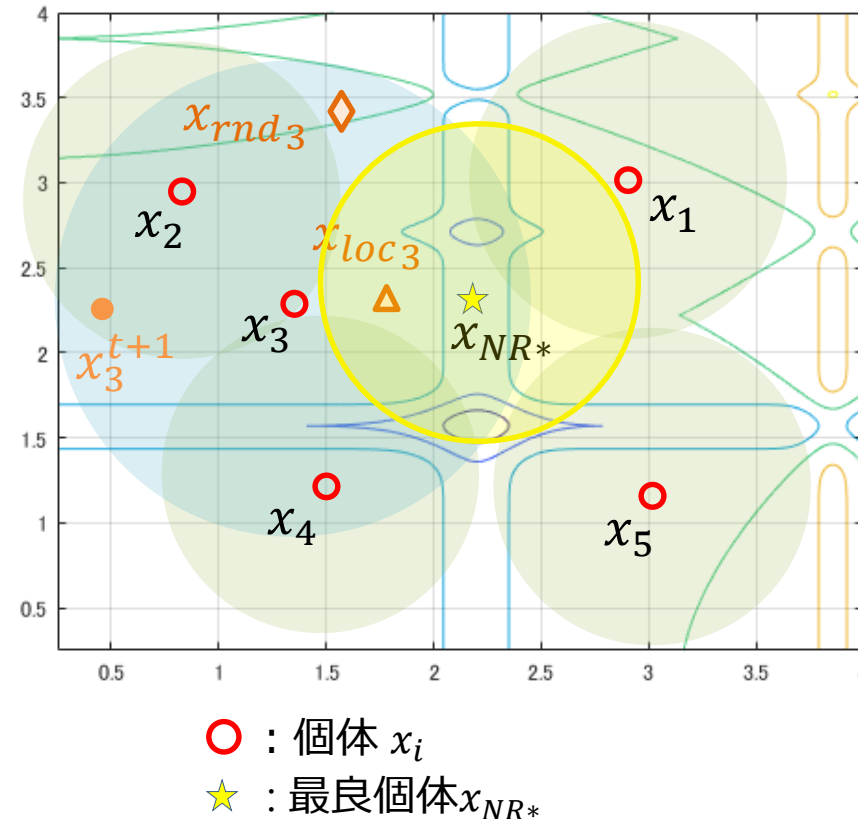
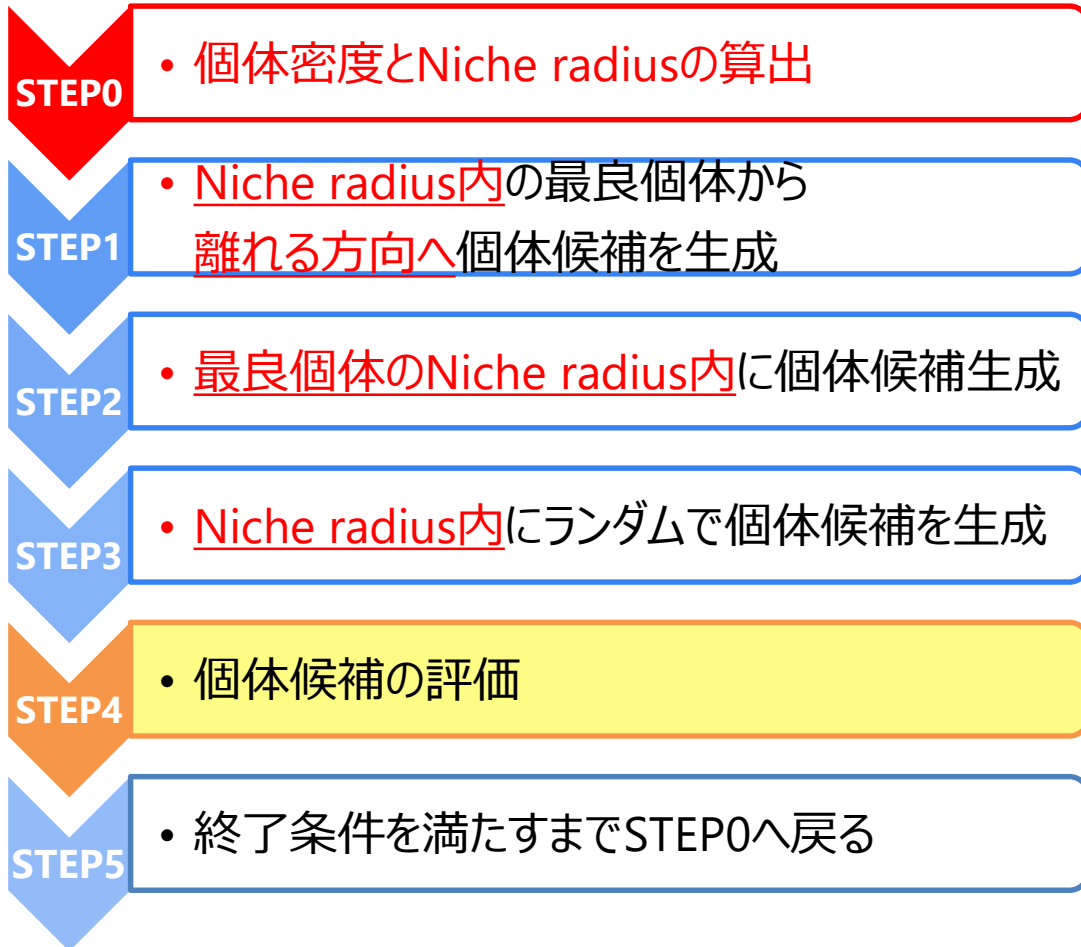
DNRBAのフローチャート

7



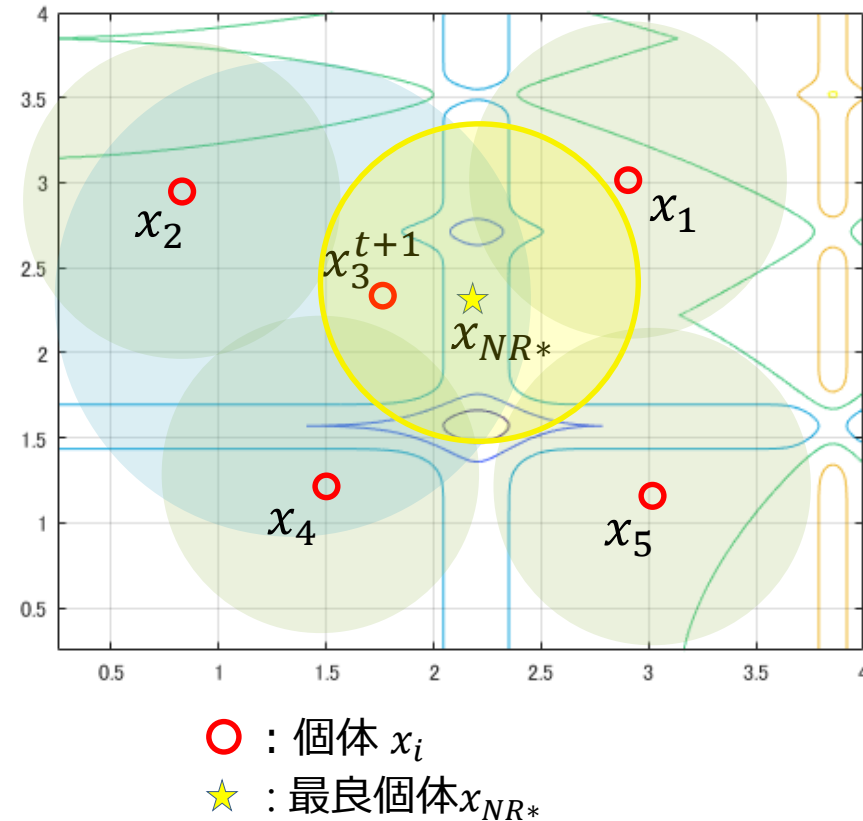
DNRBAのフローチャート

7

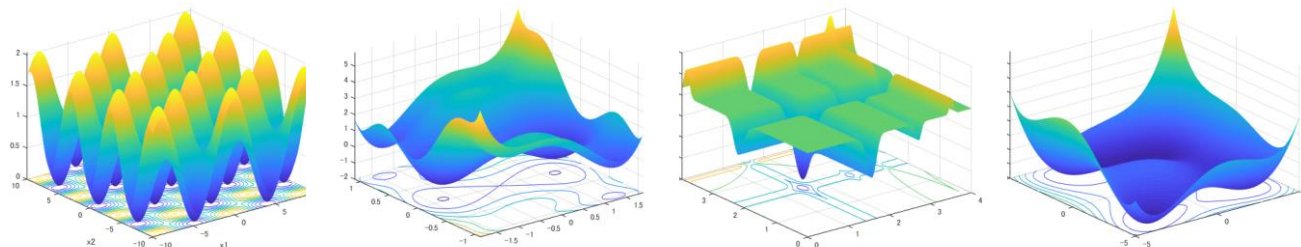


DNRBAのフローチャート

7



複数解(最適解と局所解)を持つ多峰性関数の最小化問題



関数	F_1 : Griewank	F_2 : Six-Hump Camel	F_3 : Michalewicz	F_4 : Himmelblau
探索範囲	$x_i \in [-10, 10]$	$x_1 \in [-2, 2],$ $x_2 \in [-1, 1]$	$x_i \in [0, 4]$	$x_i \in [-5, 5]$
最適解数 / 局所解数	1 / 16 (17)	2 / 2 (4)	1 / 1 (2)	4 / 0 (4)
カッコ内はピークの数 (最適解と局所解の合計)				

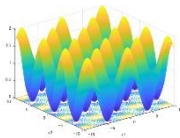
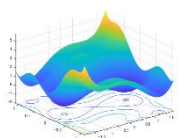
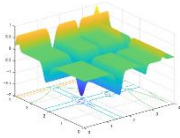
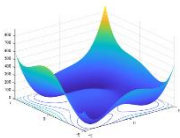
評価指標 $PR = \frac{\sum_{run=1}^{MR} \text{各試行におけるアルゴリズムが発見したピーク数}}{\text{全てのピーク数} * \text{試行回数}} \quad [X.Li, et. al, 2013]$

各ピークの座標とその最近傍個体座標の差分が閾値 ε 未満であれば、そのピークを発見したと定義

パラメータ設定

個体数: N	50
世代数: Generation	10000
次元数: D	2
試行回数: MR	30
周波数: f_{min}	0
周波数: f_{max}	1
ラウドネス: A^0	1
パルスレート: r^0	rand [0, 1]
α	0.9
γ	0.9
ε	0.1

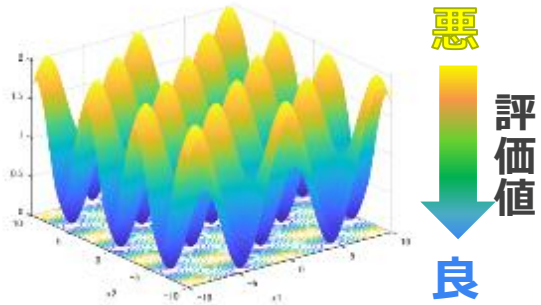
PR値と標準偏差(試行回数30試行)

	BA	NSBA	NRBA	DNRBA
Function	Mean \pm SD	Mean \pm SD	Mean \pm SD	Mean \pm SD
F_1 	0.059 \pm 0	0.371 \pm 0.041	0.692 \pm 0.098	0.745 \pm 0.061
F_2 	0.492 \pm 0	0.50 \pm 0	0.992 \pm 0.045	1 \pm 0
F_3 	0.50 \pm 0	0.50 \pm 0	0.70 \pm 0.245	1 \pm 0
F_4 	0.808 \pm 0.108	1 \pm 0	0.858 \pm 0.124	0.867 \pm 0.127

F_1 から F_3 関数までDNRBAの探索性能が最も高い

結果 - 最終世代における個体の分布

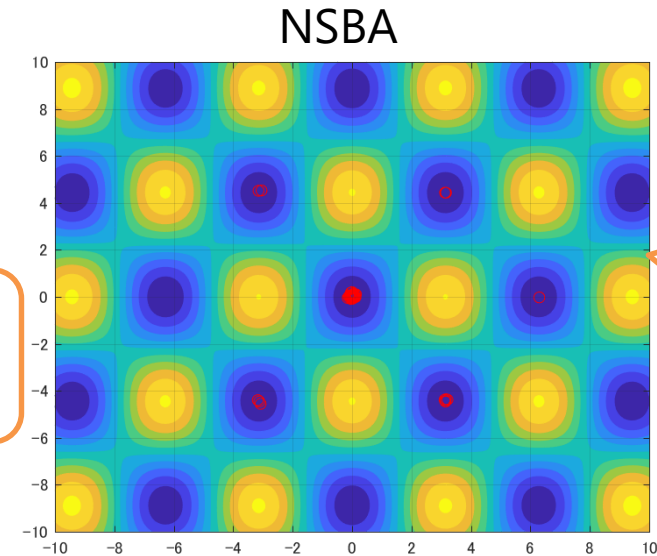
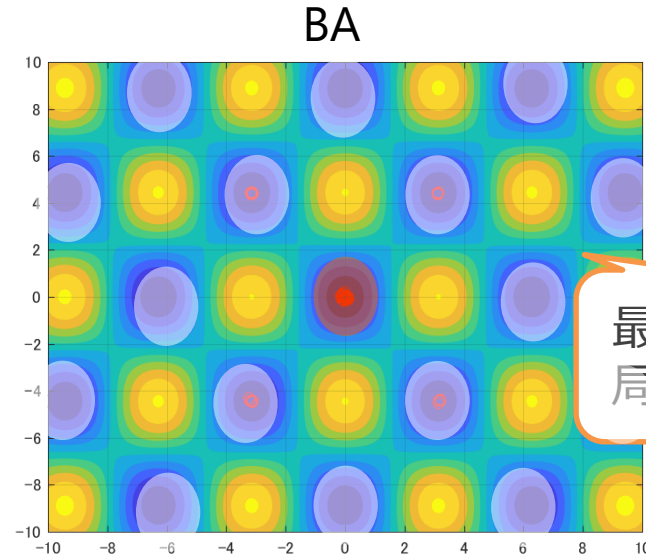
10



F_1 : Griewank

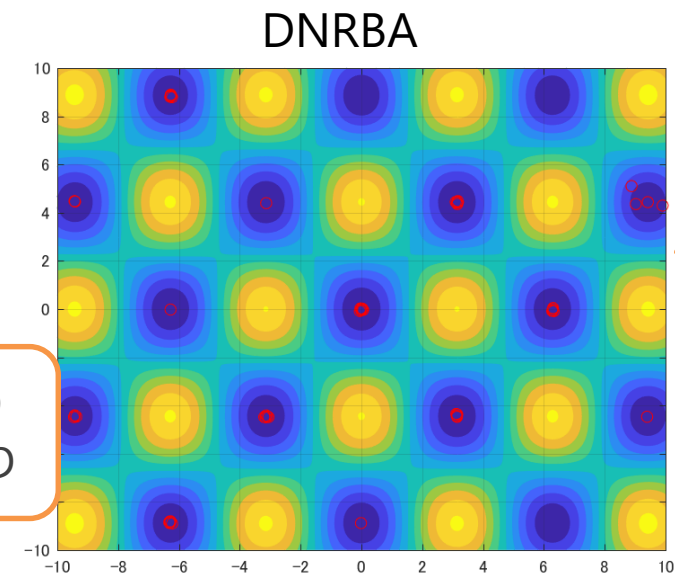
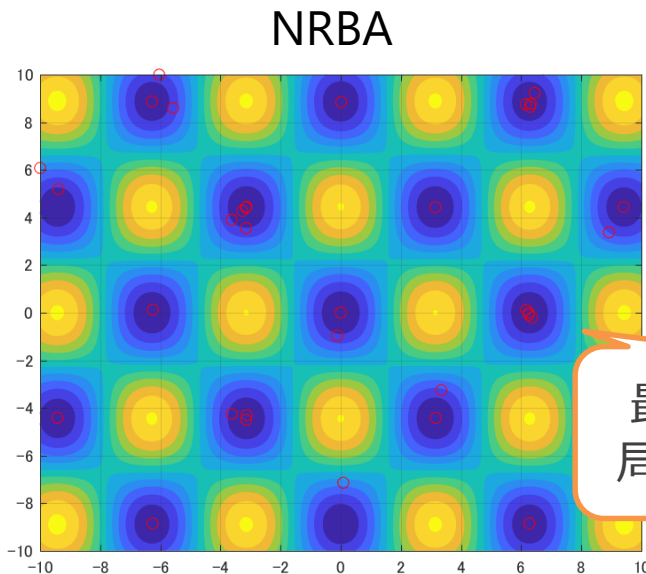
最適解及びその周辺
の局所解を捕捉

多くの局所解を捕捉



最適解1つ
局所解5つ

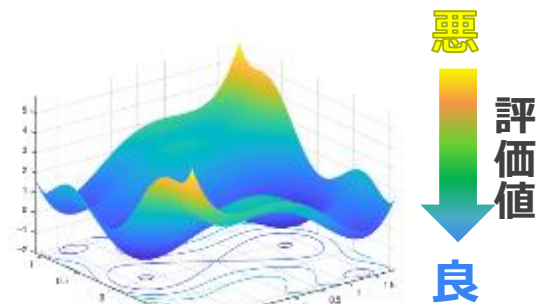
○: 個体



最適解1つ
局所解13つ

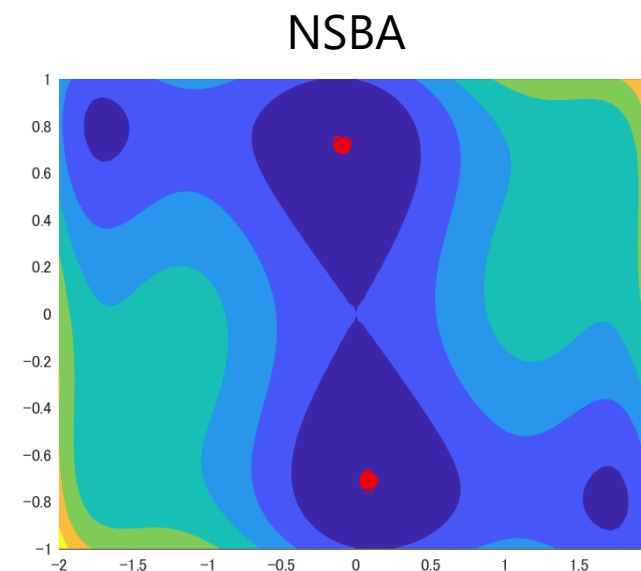
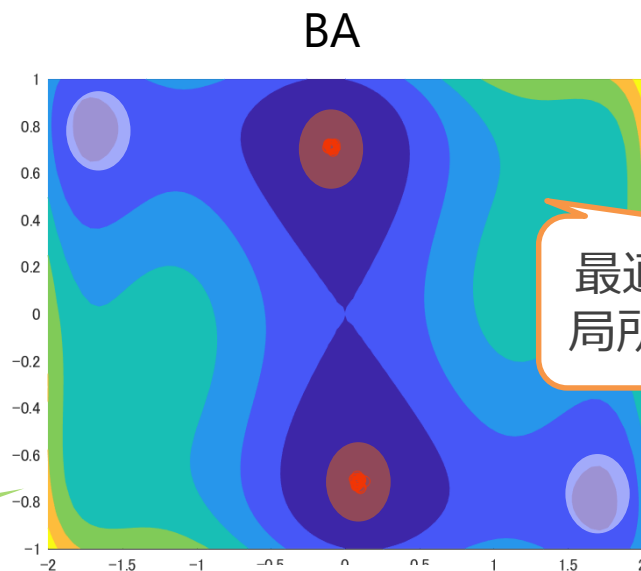
結果 - 最終世代における個体の分布

11



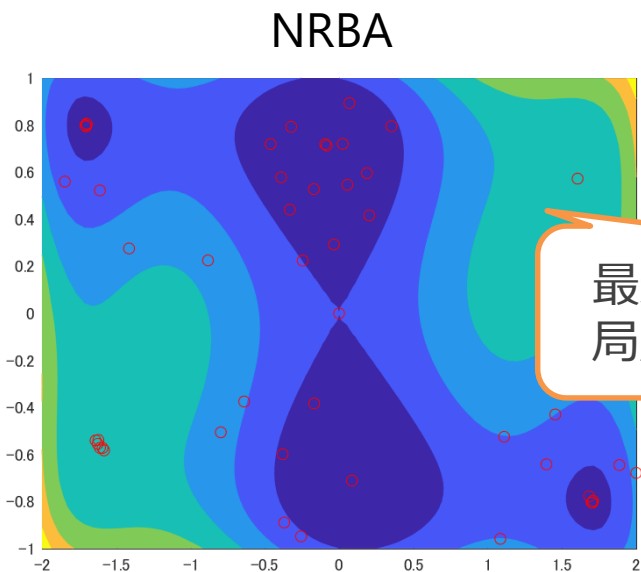
F_2 : Six
- Hump Camel Back

局所解を捕捉できない

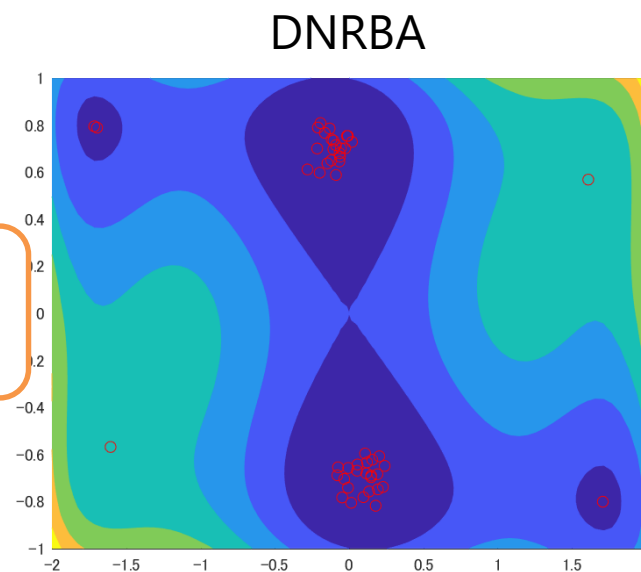


最適解2つ
局所解なし

○: 個体



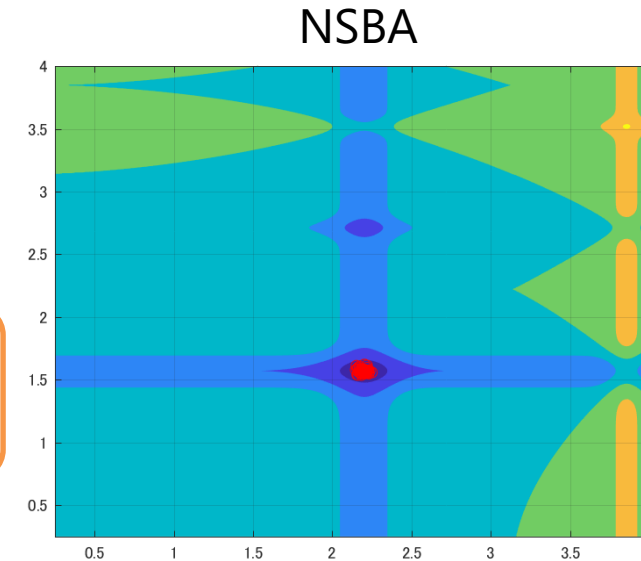
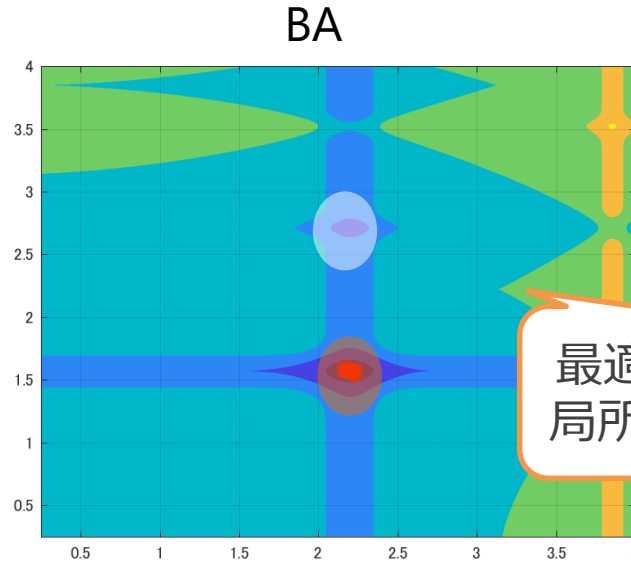
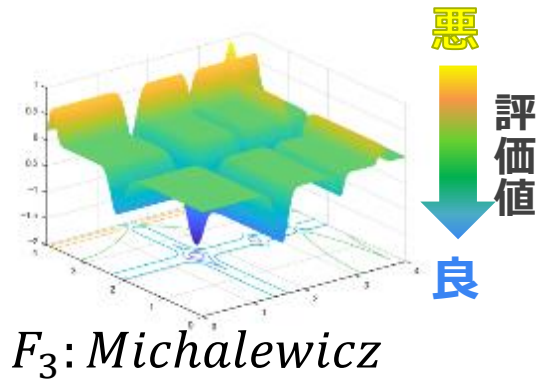
最適解2つ
局所解2つ



最適解2つ
局所解2つ

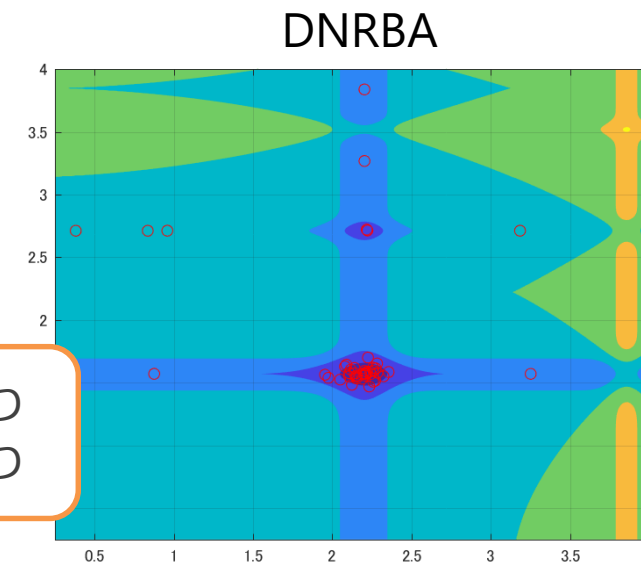
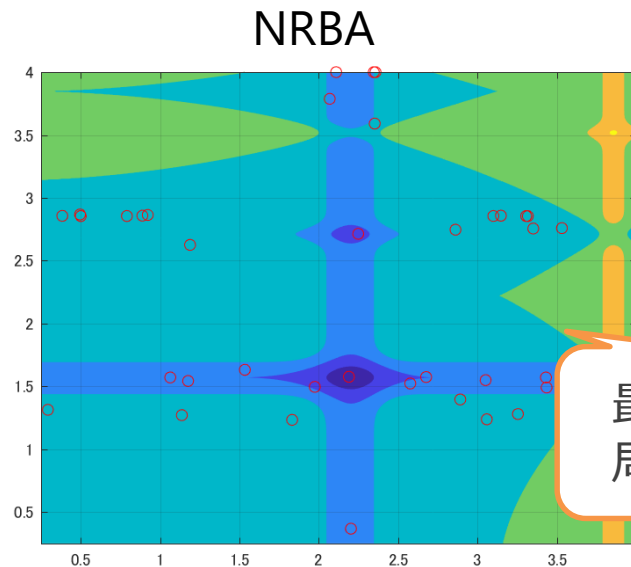
結果 - 最終世代における個体の分布

12



最適解1つ
局所解なし

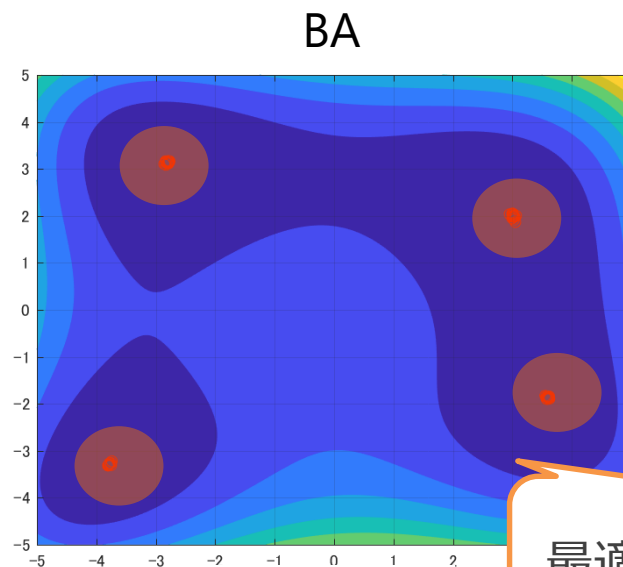
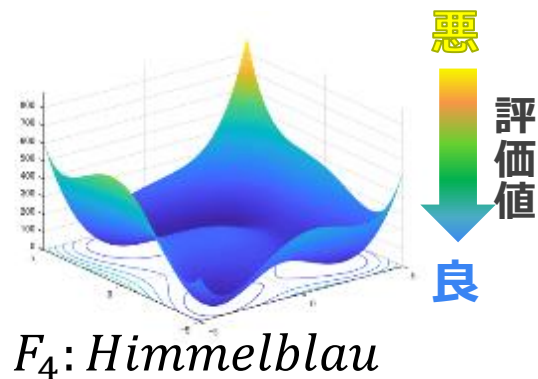
○: 個体



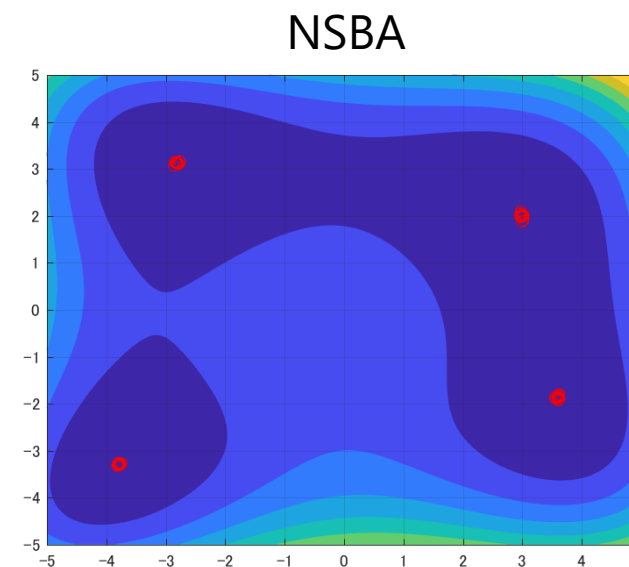
最適解1つ
局所解1つ

結果 - 最終世代における個体の分布

13

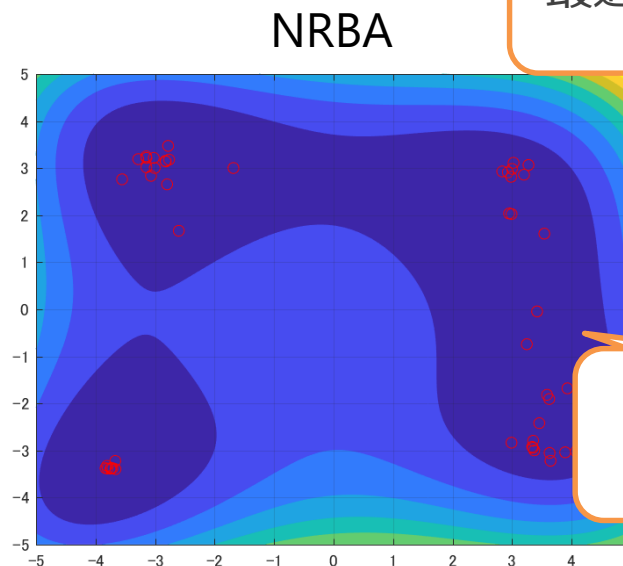


最適解4つ

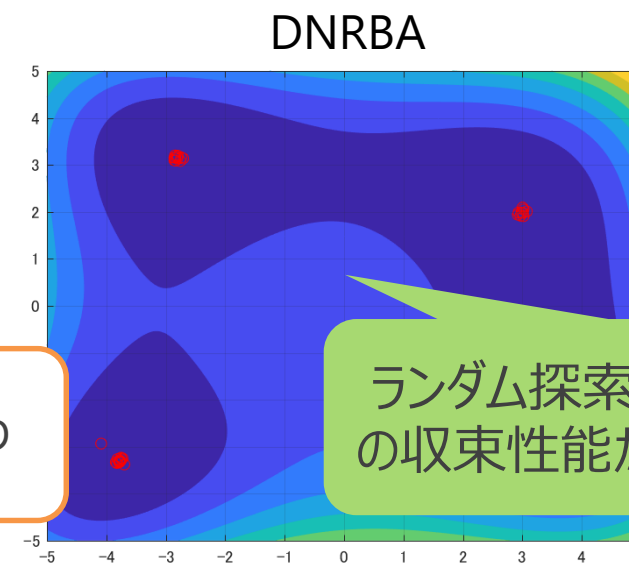


最適解4つ

○: 個体



最適解4つ



最適解4つ

ランダム探索による最適解への収束性能が従来よりも弱い

目的

複数解(最適解 / 局所解)を同時に探索可能なアルゴリズムの構築

提案

個体密度によって, 動的に探索領域を決定する
Dynamic Niche Radiusを用いた探索アルゴリズム

- Niche radius内の最良個体から遠ざかる探索
- Niche radius内の最良個体付近を局所探索
- 各個体のNiche radius内をランダム探索

知見

複数解(最適解 / 局所解)を持つ $F_1 \sim F_3$ 関数において探索性能が向上
最適解のみを持つ F_4 関数においてNSBAが有効

今後の課題

他の複数解探索手法との性能比較