

人工蜂コロニーアルゴリズム改良のための 島モデル探索手法

日大生産工 小林 聖哉 日大生産工 山内 ゆかり

1. まえがき

近年の社会で必要とされるシステムの高度化、複雑化に伴い、大規模最適化問題の解決策として、実用的な計算コストと優良解を派遣可能ということで、進化的計算手法が注目されている。進化的計算手法とは、生物の進化のメカニズムを最適化問題のモデルにした最適化問題の解法である。進化的計算には、主にダーウィンの進化論がモデルの遺伝的アルゴリズムや、生物の集団行動をモデルにした群知能がある。

近年では群知能の中でも特に、ミツバチの群れの採餌行動をモデルにした人工蜂コロニー (Artificial Bee Colony : ABC) アルゴリズムが注目されている。ABCアルゴリズムは、関数最適化を目的として作られたアルゴリズムで、高次元問題や短期性、多岐性などの様々な性質の問題に対して有効であることが検証されている。

しかし、ABCアルゴリズムは子個体の多様性を重視した探索を行う結果、優良解に収束するまでに多くの世代数を必要とする。この欠点を補うために、ハイブリット手法が用いられ、算術交叉型ABC (Arithmetic Crossover based Artificial Bee Colony: AC-ABC) アルゴリズム [1] や大域探索型ABC (Global Search type Artificial Bee Colony: GS-ABC) アルゴリズム [1] など提案されている。

2. ABCアルゴリズム

ABCアルゴリズムはD.Karadogaらにより2005年に提案された軍知能の一つである。蜜蜂の働き蜂、見物蜂、偵察蜂をモデルにした、3種の探索を行う [2]。以下にABCアルゴリズムの探索法を示す。

Step 0 初期化

世代数 (以下、 g とする) を1とし、問題の定義域内に初期個体をランダムに作成する。個体数を N としたとき、働き蜂 (以下、 N_e とする)、見物蜂 (以下、 N_a とする) の数は $N=N_e=N_a$ となる。

Step1 働き蜂

各個体に付き1回ずつ式 (1) を用いて、ランダムに選択された個体内の1つの変数成分を更新する。また更新した個体を、全個体 $x_i (i=1, 2, \dots, N)$ とは別に、更新候補隊 $v_i (i=1, 2, \dots, N)$ を作成する。次に x_i と v_i をグリーディ選択 (2) を行い、個体を更新する。

$$v_{ij}^g = x_{ij}^g + \phi_{ij}^g (x_{ij}^g - x_{mj}^g) \quad (1)$$

$$x_i^g = \begin{cases} v_i^g & \text{if } f_{v_i^g} > f_{x_i^g} \text{ then } TC_i = 0 \\ x_i^g & \text{others then } TC_i = TC_i + 1 \end{cases} \quad (2)$$

上式において、 i は個体番号、 m は i 以外のランダムに選択された個体番号、 ϕ は-1~1までの一葉乱数、 f_i は個体 i の適応度を示す。

Step 2 見物蜂

式 (3) を用いて計算した個体を用いて計算した個体の相対価値確立 P_i^g に基づくルーレット選択により1つの個体 f_i^g を選択する。その個体を式 (1) を用いて計算し式 (2) を用いて更新する。

$$P_i^g = \frac{f_i^g}{\sum_{n=1}^N f_n^g} \quad (3)$$

Step3 偵察蜂

$TC \geq \text{limit}$ を満たす個体を式 (4) に置き換える。

$$x_{ij} = x_{max} + r_{ij}^g (x_{max} - x_{min}) \text{ if } TC_i \geq \text{limit} \quad (4)$$

x_{max} は定義域の最大値、 x_{min} は定義域の最小値 r は0~1までの一葉乱数を表す。置き換えられた場合その個体の TC は初期化される。

Step4 裁量個体

その世代の探索を一通り終えたら、全個体の適応値を求め最良の適応値を上回った場合、最良を更新する。

Step5 終了条件

終了条件を満たしている場合、探索を終了する。満たしていない、 $g=g+1$ を行いStep1に戻る。

ABCアルゴリズムは以上のプロセスを用い行われる。

3. 従来手法

3.1 AC-ABC

ABCアルゴリズムのハイブリット手法として算術交差型ABCアルゴリズム (AC-ABC) と大域交叉型ABCアルゴリズム (GS-ABC) がある。

AC-ABCアルゴリズムは、ABCアルゴリズムの追従蜂フェイズにGAの交叉の要素を取り入れたABCアルゴリズムである。追従蜂フェイズの式 (1) の代わりに式 (5) を用いたものである。更新式に関しては (2) の式を用いる。

$$v_{ij}^g = \begin{cases} \lambda x_{ij}^g + (1 - \lambda) y_{ij}^g & \text{if } r_{ij}^g \leq CR \\ x_i^g & \text{others} \end{cases} \quad (5)$$

y_{ij}^g はルーレット選択によってえられた個体である。CRは交差率である。

AC-ABCのメリットは、優良解の探索能力を持ちながら、多様性も同時に保っている点である。デメリットは、GAの交叉の要素を組み込んだ結果ABCアルゴリズムと比べ、多峰性問題で局所解に陥る確率が高い点である。

3.2 GS-ABC

GS-ABCアルゴリズムはABCアルゴリズムの収穫蜂・追従蜂フェイズにGAの突然変異の要素を組み込んだもの。このGS-ABCアルゴリズムは、突然変異確立を組み込むが基本の式は元のABCアルゴリズムを用いているのが特徴。GS-ABCアルゴリズムのイメージを図 (1) に示す。

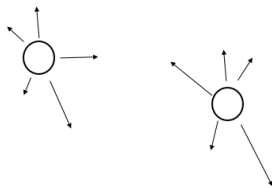


Fig.1 GS-ABC のイメージ

収穫蜂・追従蜂の式 (1) の代わりに式 (6) を用いる。更新式は (2) を用いる。

$$v_{ij}^g = \begin{cases} x_{ij}^g + \phi_{ij}^g (x_{ij}^g - x_{mj}^g) & \text{if } r_{ij}^g \leq \alpha \\ x_i^g & \text{others} \end{cases} \quad (6)$$

α は突然変異率である。

GA-ABCアルゴリズムのメリットは、それぞれの個体が毎世代、毎回周囲を大量に探索を行っているので局所解に陥ることがなく、早い段階で最適解に至ることができる。デメリットは、毎世代毎個体毎次元で探索を行うのでABCア

ルゴリズムと比べ、探索に膨大な時間がかかる事である。

4.1 提案手法

従来手法のAC-ABCは遺伝的アルゴリズムの算術交差をABCに取り入れたものであるが、デメリットに局所解に陥ってしまうことが多々起きてしまうことがある。

遺伝的アルゴリズムでは局所解に陥る対策として提案された、並列分散GA[3]などが有名である。これは、合計の集団の個体をそれぞれいくつかの島に分け、その島の中で遺伝的処理を行うものである。また、移住という要素もあり、これは島ごとの個体を移動させるものである。

本研究では、この並列分散の要素をAC-ABCアルゴリズムに組み込んだ並列分散算術交差型人工蜂コロニーアルゴリズム (Distribution Arithmetic Crossover based artificial Bee Colony algorithm: PDAC-ABC) を提案する。

4.2 DAC-ABC

ここでは、本論文の第一の提案手法である並列分散算術交差型ABCアルゴリズム (PDAC-ABC) について説明する。並列分散は集団の中の個体をいくつかに分け島ごとでアルゴリズムの処理を行うものである。局所解対策に用いられる手法である。島モデルともいわれている。算術交差は実数地であることを生かした交叉の一手法である。

アルゴリズムのフローチャートを図 2 示す。

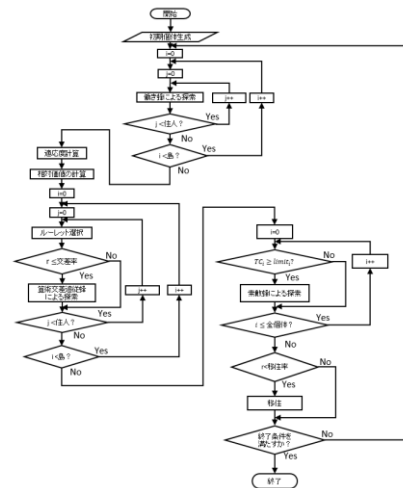


Fig.2 DAC-ABC のフローチャート

本アルゴリズムでは、さらに探索性能を上げるために追従蜂フェイズの探索にGS-ABCと同じように、ルーレット選択で選ばれた個体の各要素ごとに交叉を行った結果を逐次適応度計算を行い式 2 を行う。GS-ABCの要素を組み込んだAC-ABC のイメージを図3に示す。

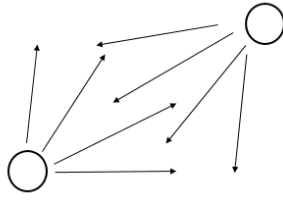


Fig.3GS-ABC の要素を組み込んだイメージ

後述するシミュレーション結果に関して、上記の機能を有り無しの結果も求める。また、比較のためAC-ABCにも同様に上記の機能有り無しの結果も求める。

5. 実験方法および測定方法

本研究の提案手法の有用性を関数最適化シミュレーションを以下に示す。まず実験方法について示す。

5.1シミュレーション条件

本研究の対象問題として、関数最適化問題のベンチマークとして知られている6種類の関数を用いて、粒子群最適化 (PSO)、ABCアルゴリズム、AC-ABCアルゴリズム、GS-ABCアルゴリズム提案手法の比較シミュレーションを行った。シミュレーションに用いた関数を式 (7) ~ (12) に示す。

Sphere関数

$$F_{\text{Sphere}}(x) = \sum_{i=1}^n (x_i^2) \quad (7)$$

Rosenbrock関数

$$F_{\text{Rosenbrock}}(x) = \sum_{i=1}^{n-1} \{100(x_{n+1} - x_i^2)^2 + (1 - x_n)^2\} \quad (8)$$

Rastrigin関数

$$F_{\text{Rastrigin}}(x) = 10 + \sum_{i=1}^n (x_i^2) \quad (9)$$

Griewank関数

$$F_{\text{Griewank}}(x) = 1 + \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \left(\frac{x_i}{\sqrt{i}}\right) \quad (10)$$

Alpine関数

$$F_{\text{Alpine}}(x) = \sum_{i=1}^n |x_i \sin(x_i) + 0.1x_i| \quad (11)$$

2^n minima関数

$$F_{2^n \text{ minima}}(x) = \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i) \quad (12)$$

6. 実験結果および検討

PSO、ABC、AC-ABC、GS-ABC、DAC-ABCアルゴリズムを用いて表1のパラメータで行った関数の実験に用いた各アル

ゴリズムのパラメータを表1に、最適化シミュレーションの結果を図4~9に示す。

表1 各アルゴリズムのパラメータ

	PSO	ABC	AC-ABC	AC-ABC-GS	GS-ABC	DAC-ABC	DAC-ABC-GS
世代数	1000	1000	1000	1000	1000	1000	1000
個体数N	100	100	100	100	100	100	100
次元数D	40	40	40	40	40	40	40
島数	—	—	—	—	—	4	4
島個体数	—	—	—	—	—	25	25
交叉率	—	—	0.5	0.5	—	0.3/0.5/0.7/0.8	0.3/0.5/0.7/0.8
減衰係数α	0.6	—	—	—	—	—	—
c_1	0.8	—	—	—	—	—	—
c_2	0.1	—	—	—	—	—	—
limit	—	N・D	N・D	N・D	N・D	N・D	N・D
λ	—	—	0.1	0.1	—	0.1	0.1
α	—	—	—	—	0.7	—	—

グラフの横軸は世代数、グラフの縦軸はそれぞれの関数を乱数シードを変更して行った20試行の最大適応度の平均値である。またグラフの最大適応度の値は0.0~1.0に正規化したものである。

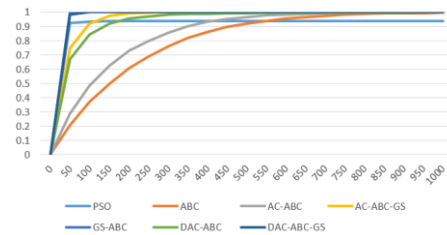


Fig.4 Sphere関数

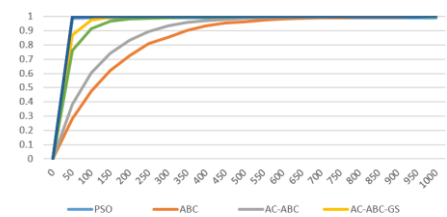


Fig.5 Rosenbrock関数

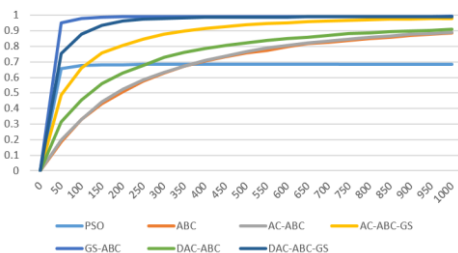


Fig.6 Rastrigin関数

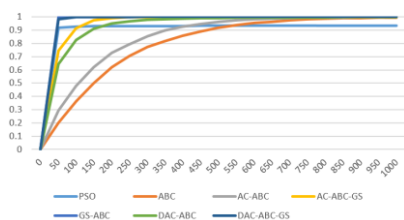


Fig.7 Griewank 関数

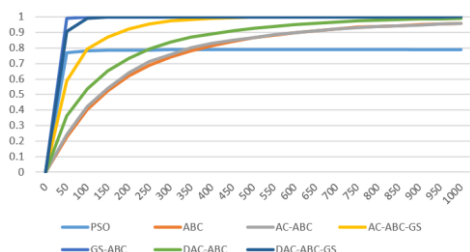


Fig.8 Alpine関数

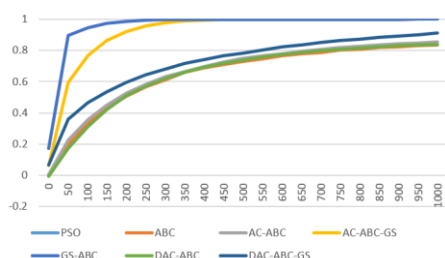


Fig.9 2^nminima 関数

図4では、DAC-ABCがAC-ABC、GS-ABCの要素を組み込んだAC-ABC (AC-ABC-ON) よりも早い段階で最適解に到達した。また、GS-ABCの要素を組み込んだ提案手法 (DAC-ABC-ON) に至ってはGS-ABCに差し迫る勢いで最適解に到達した。図5でも同様に従来研究よりも良い結果が求められた。

図6では従来研究よりも精度は上がったが残念ながら1000世代では最適解に到達できなかった。だがDAC-ABC-GSは唯一最適解に至ることができた。図7は図4・5同様に従来研究より良い結果が求まった。

図8では、DAC-ABCはAC-ABCより良い結果を得たがAC-ABC-GSを超える精度は得られなかった。図9では、残念ながらDAC-ABCはAC-ABCとさほど変わらない結果が得られた。

7. 考察

シミュレーション結果からDAC-ABCの性能の向上は見られたがいくつかの関数では最適解に至れない結果となった関数がある。

世代ごとのデータを見てみると、これはある程度世代が進み大抵の個体が近似解周辺に集まることによって、ルーレット選択が機能していないことが原因と考えられる。

次に本研究のアルゴリズムの処理速度について検討する。大きな数字で検討するため、各アルゴリズム20試行の時間の合計を表2に示す。

表 2 各特性の比較

#	PSO	ABC	AC-ABC	AC-ABC-GS	GS-ABC	DAC-ABC	DAC-ABC-GS
Sphere	15	6	6	8	147	4	18
Rosenbrock	19	14	14	20	361	9	44
Rastrigin	20	12	12	16	283	8	34
Griewank	73	26	14	21	385	9	46
AlFine	16	6	6	8	304	6	17
2nMinims	23	21	21	28	534	13	63

実行にかかる時間を見て、処理速度は従来手法とさほど変わらず、場合によってはさらに早く動く結果を得られた。DAC-ABC-GSも高性能なGS-ABCと比べかなり早い時間でよい結果を求めることができた。GS-ABCは高い探索の代わり探索時間が膨大にかかってしまうが追従蜂フェイズなどの特定の個体を探索する際に組み込むレベルなら、さほど時間がかからない結果が出た。

8. まとめ

本研究ではDAC-ABCの提案手法で従来手法であるABC、AC-ABC、GS-ABCアルゴリズムと比較を行った。結果は、最適解到達世代数ではGS-ABCに及ばなかったもののABC、AC-ABCアルゴリズムよりも優れた探索性能が確認された。また、プログラム所要時間はさほど変わらないことから、優良解への収束性能を高めることができれば、最適化問題に対する優良な解法と期待できる。

今後の課題としては、島ごとのパラメータの設定や単峰性弱い多峰性問題に対してある程度個体が近似解に収束した際、ルーレット選択が余り意味をなしていないので、有用な選択方法の検証、または、GS-ABCの要素を組み込むなど検証が必要であると考えられる。

参考文献

- 1) 前田陽一郎、人工蜂コロニーアルゴリズムのためのハイブリッド探索手法、知能と情報 (日本知能情報ファジィ学会誌) Vol.30, No.2, pp.556-563(2018)
- 2) 大谷紀子、「進化計算アルゴリズム入門 生物の行動科学から導く最適解」オーム社 p81-106.
- 3) Tanase, R., "Distributed Genetic Algorithms," Proc. 3rd Int. Conf. Genetic Algorithms, (1989), 434