

# GABC アルゴリズムの拡張とその実験的評価

近藤 久†

茨城大学工学部知能システム工学科†

## 1 はじめに

近年, 単純な知能のみを有する個体が群れを成すことによって非常に高度な知能が創発される群知能 (*swarm intelligence*) [1] が注目され活発な研究が行われている. 代表的な群知能のひとつとして ABC (*artificial bee colony*) アルゴリズム (以後, ABC と略記) がある [2, 3]. ABC は蜜蜂の採餌行動を模倣することによって最適化問題を解く. 文献 [4] において ABC の性能は GA や PSO (*particle swarm optimization*) と互角がそれ以上であることが示されている. 最近, Zhu 等は PSO の更新式に基づき群れ全体の最良解を用いて ABC の解探索を誘導するように改良した GABC (*gbest-guided ABC*) を提案した [5].

本稿では, GABC の更なる性能改善のための拡張を提案する. 具体的には群れが探索した現時点の最良解の適応度が, ある条件のもとでこれまでの群れ全体の最良解の適応度よりも低くなった場合, 最低適応度を持つ解とこれまでの最良解を置換する. この置換によって GABC よりも性能が改善されることを実験を通して確認した.

## 2 ABC と GABC

### 2.1 ABC[3]

自然界においては, 一般に餌場から蜜を集める 3 種類の蜜蜂がいる. 収穫蜂 (*employed bee*), 追従蜂 (*onlooker bee*) と偵察蜂 (*scout bee*) である. ABC では, 収穫蜂の数を  $n$  とすると追従蜂の数も  $n$  であり, 巣を構成する蜜蜂の総数 (コロニーサイズ) は  $2n$  となる. 餌場 (候補解) の数は収穫蜂の数と同数である. ABC の枠組を図 1 に示す.

```

while 停止条件を満たさない do
  収穫蜂フェーズ: 各解の更新を行う.
  追従蜂フェーズ: 各解の適応度に基づいて解を
                  選択し, その解の更新を行う.
  偵察蜂フェーズ: 設定された回数の間に一度も
                  更新されなかった解の 1 つを
                  ランダムに生成した新たな解
                  と置換する.
end

```

図 1 ABC アルゴリズムの枠組

$x_i (i = 1, \dots, n)$  を  $D$  次元の実数値ベクトル,  $x_{ij}$  (または  $v_{ij}$ ) をベクトル  $x_i$  (または  $v_i$ ) の  $j (j = 1, \dots, D)$  番目の成分をあらわすとする. 収穫蜂フェーズでは解  $x_i$  を用いて次の更新式により更新する ( $v_i = x_i$  とする).

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (1)$$

ここで,  $j$  はランダムに選択される.  $x_k$  (ただし  $k \neq i$ ) はランダムに選択された  $x_i$  以外の解である.  $\phi_{ij}$  は区間  $[-1, 1]$  中の乱数である.  $v_i$  と  $x_i$  の適応度を比較し, もし  $v_i$  が優れていれば  $x_i$  と置き換える. さもなくば,  $x_i$  の試行カウンタを 1 増やす. 対象問題が目的関数  $f(x)$  の値を最小とする  $x$  の値を求める最小化問題ならば, 適応度は次の評価関数により求められる.

$$fit_i = \begin{cases} \frac{1}{1 + f(x_i)} & \text{if } f(x_i) \geq 0 \\ 1 + abs(f(x_i)) & \text{if } f(x_i) < 0 \end{cases} \quad (2)$$

追従蜂フェーズでは, 収穫蜂によって更新された解の適応度に基づいた確率  $p_i = fit_i / \sum_n fit_n$  を用いて解を選択し, その解に対して収穫蜂と同様の更新処理を行う. 収穫蜂フェーズとの違いは追従蜂フェーズでは適応度に基づいて選択された解のみが更新されることである.

偵察蜂フェーズでは, 収穫蜂フェーズと追従蜂フェーズを通して予め決められた打ち切り回数 ( $C_{limit}$ ) の間に一度も更新されなかった解をランダムに生成した解に置換する.

以後, 図 1 の 1 回の while ループをサイクルと記述する.

### 2.2 GABC[5]

GABC は PSO [1] の更新式に基づいて, サイクル  $t$  までに群れ全体が発見した最良解 (*global best* (*gbest* と略記)) を利用し, サイクル  $t+1$  の ABC の解探索を誘導するように改良した手法である. GABC は (1) 式の代わりに次の更新式を用いる.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \quad (3)$$

ここで,  $y_j$  は *gbest*:  $y$  の  $j$  番目の成分,  $\psi_{ij}$  は区間  $[0, C]$  中の乱数であり,  $C$  は非負の定数である.  $\psi_{ij} = 0$  の場合は ABC と同じである.

## 3 提案手法

GABC は, その性能が ABC よりも優れていることが示されているが, どちらの手法とも  $C_{limit}$  の値に大きく依存する. これは知識利用 (*exploitation*) と探索 (*exploration*) のバランスを取るパラメータであり, 解の近傍を探索する回数を決定する. 大きな値の場合は,

On an expansion of GABC algorithm and it's experimental evaluation

† Hisashi KONDO (*H.Kondo@mx.ibaraki.ac.jp*)

Department of Intelligent Systems Eng., College of Eng., Ibaraki University, 4-12-1 Naka-Narusawa, Hitachi, Ibaraki 316-8511, Japan

適応度の低い解も高い解も場合によっては同じ回数の近傍探索を行うこととなり非効率的である．小さな値の場合は見込みのありそうな解の近傍探索回数が少なすぎて最良解を発見出来ない．この値を越えた解は偵察蜂フェーズでランダムに生成された解と置換される．ランダムに生成された解は通常は適応度が低く，次々と偵察蜂フェーズで今までの探索解が置き換えられてしまうと群れ全体の多様性は保証されるがこれまでの探索結果を利用出来なくなってしまう．

提案手法では，サイクル  $t$  で偵察蜂フェーズによるランダム解への置換があった場合に，次のサイクル  $t + 1$  における収穫蜂フェーズと追従蜂フェーズ終了段階での最良解の適応度が  $g_{best}$  の適応度より低い場合に限り最低適応度の解と  $g_{best}$  を置換することにした．これは過去に収穫した質の高い餌場に一定期間をおいて再度戻ってみることに相当する．この操作を再訪問フェーズと呼ぶことにする．再訪問フェーズを取り入れることによって，早期に改善見込みの少ない解を捨て， $g_{best}$  に戻ってみることによって，その近傍探索と他の解の近傍探索に  $g_{best}$  を利用した解の改善を試みることが出来る．再訪問フェーズを取り入れた GABC を GABC+ と呼ぶことにする．

#### 4 実験結果

ABC, GABC 及び GABC+ を実現し，関数値最小化実験を行った．Rosenbrock 関数  $f_1(\vec{x})$  (4) 式，次元  $D = 5$ ,  $\vec{x}$  の初期範囲  $[-50, 50]^D$ , 最小解  $\vec{x}^* = [1, 1, \dots, 1]$ ,  $f_1(\vec{x}^*) = 0$ , Rastrigin 関数  $f_2(\vec{x})$  (5) 式，次元  $D = 100$ ,  $\vec{x}$  の初期範囲  $[-5.12, 5.12]^D$ , 最小解  $\vec{x}^* = [0, 0, \dots, 0]$ ,  $f_2(\vec{x}^*) = 0$  の 2 つの実験結果について述べる (その他の関数については大会当日に述べる)．

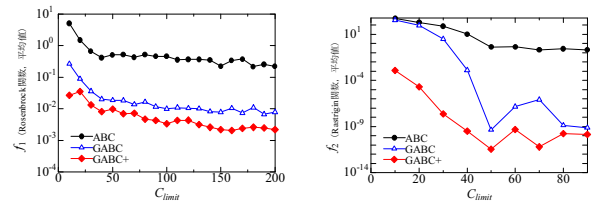
$$f_1(\vec{x}) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \quad (4)$$

$$f_2(\vec{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (5)$$

アルゴリズムのパラメータは文献 [5] と同様にコロニーサイズ 80 (収穫蜂 40, 追従蜂 40), 繰り返し回数 5000 とし, 各々 30 回の実験を行った． $C_{limit}$  の値の変化に対する関数値 (平均値) の変化を図 2 に示す (片対数となっていることに注意)．

図 2 より, どちらの関数においても GABC+ の性能が優れていることがわかる．関数  $f_1$  では  $C_{limit} = 160$ , 関数  $f_2$  では  $C_{limit} = 50$  のときに, それぞれ最小値となった．関数  $f_1$  では  $C_{limit}$  の変化に対して関数値の変化があまりないが, 関数  $f_2$  では大きく変化していることがわかる．

$C_{limit} = 50$  のとき GABC+ における 30 回の実験の中で関数  $f_2$  が最小値 ( $3.38E-14$ ) になった回と同じ乱数シードを用いた場合のサイクル  $t$  における  $g_{best}$  での関数値の変化を図 3 に示す．ABC では関数値は  $t = 3500$



(a) Rosenbrock 関数 (b) Rastrigin 関数

図 2 関数値 (平均値) の変化

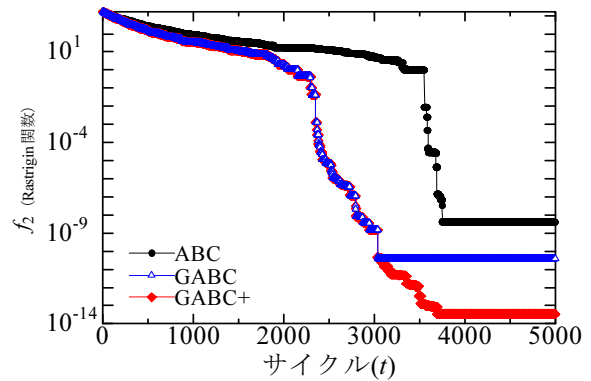


図 3 Rastrigin 関数の  $g_{best}$  での値の変化

過ぎまでゆっくりと低下している． $t = 3756$  で最小値 ( $4.02E-09$ ) となり収束した．GABC と GABC+ は  $t = 3000$  過ぎまで同じ振舞いを示しながら降下し, GABC では  $t = 3043$  で最小値 ( $4.05E-11$ ) となり収束した．GABC+ では, その後も関数値は降下し,  $t = 3699$  で最小値 ( $3.38E-14$ ) となった．

#### 5 まとめと今後の課題

本稿では, GABC アルゴリズムの拡張とその実験結果について述べた．本拡張手法の考えは非常に単純であるが実験を通してその有効性を示した．今後の課題として, (1) さらなる実験比較と (2)  $g_{best}$  と最低適応度解の置換タイミングの検討などが挙げられる．

#### 参考文献

- [1] Kennedy, J. and Eberhart, R. C.: Swarm Intelligence, Morgan Kaufmann Publishers(2001).
- [2] Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Erciyes University, Kayseri, Turkey, Technical Report-TR06(2005).
- [3] Karaboga, D. and Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony(ABC) algorithm, J. of Global Optimization, Vol. 39, No. 3, pp.459-471(2007).
- [4] Karaboga, D. and Akay, B.: A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation, Vol. 214, Issue 1, pp.108-132(2009).
- [5] Zhu, G. and Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation, Vol. 217, Issue 7, pp.3166-3173(2010).