

Artificial Bee Colony における学習パラメータの影響に関する検討 Study on Influence of Parameter Settings in Artificial Bee Colony Algorithm

竹内 広樹[†] 荒川 正幹[†]
Hiroki Takeuchi Masamoto Arakawa

1. はじめに

近年、最適化問題を解く手法として群知能に注目が集まっている。最適化問題は、関数 $y=f(x)$ の y が最適な値になるような x の値を求める問題である。最適化問題において大域的最適解を求めることは困難であり、許容できる計算時間内に大域的最適解を求めることを保証した手法は存在しない。このような背景から現実的な計算時間で近似解を探索することが可能な群知能が考案されている。

代表的な群知能としては Ant Colony Optimization (ACO)、Particle Swarm Optimization (PSO)、Artificial Bee Colony (ABC)[1] アルゴリズムなどを挙げることができる。なかでも ABC アルゴリズムは高次元最適化問題に対する解探索性能が優れていることが報告されている。また、ABC アルゴリズムはこれまでに多くの改良が行われており、その中でも Gbest-guided artificial bee colony (GABC) [2] アルゴリズムは解探索性能が優れている。しかし、各パラメータの探索性能への影響については、解明されていない点が多い。

本研究の目的は、ABC の学習パラメータ (1 度に変更する変数の数) が探索にどのような影響を与えるかを調査することである。ABC および GABC について、4 つのベンチマーク関数を対象としたシミュレーションを実施し、パラメータの影響を検討するとともに、探索性能の向上を図る。

2. 手法

2.1 ABC アルゴリズム

ABC アルゴリズムとは、ミツバチの群れによる採餌行動にヒントを得て考案された最適化アルゴリズムである。解候補を食物源に見立て、それをコロニー内のミツバチが繰り返し探索する。コロニーは雇用ハチ、見物ハチ、偵察ハチによって構成され、雇用ハチは、各自が担当する食物源の近傍を探索する。担当する食物源が枯渇した場合、その雇用ハチは偵察ハチとなる。見物ハチはランダムに選択した食物源の近傍を探索する。食物源の適合度が高いほど、高確率で見物ハチに選択される。制御パラメータとして、雇用ハチの数、最大サイクル数、近傍の探索を断念するまでの試行回数が設定可能である。

ABC では、全ての食物源が初期化された後、探索サイクルが繰り返される。雇用ハチおよび見物ハチによる近傍の探索は、次式にしたがって行われる。

$$v_{ij} = x_{ij} + rand[-1,1] \cdot (x_{ij} - x_{kj}) \quad (1)$$

ここで、 $rand[-1,1]$ は $-1 \sim 1$ の範囲の一様乱数、 j はランダムに選ばれた 1 つの次元である。 v_i は新たな解候補、 x_i は現在の解候補、 x_k はランダムに選択された解候補である。

適合度 $f(v_i)$ を求め、 $f(v_i)$ が $f(x_i)$ より大きい場合、 x_i を v_i で更新する。

2.2 GABC アルゴリズム

GABC アルゴリズムは、Zhu らによって提案された ABC の改良アルゴリズムである。これまで発見された中で最も高い適合度を持つ食物源の情報を、新しい解候補の生成に用いる。新たな解候補 v_i は次式によって計算される。

$$v_{ij} = x_{ij} + rand[-1,1] \cdot (x_{ij} - x_{kj}) + rand[0,C] \cdot (y_j - x_{ij}) \quad (2)$$

ここで y は、これまで発見した中で最も高い適合度を示す解候補である。 C は任意の定数であり、本研究においては 1.5 とした。

3. 実験

ABC、GABC は 1 度の探索で x_i の 1 つの変数しか変更しない。そこで複数の変数を変更した場合、探索性能にどのような影響を与えるかを調査した。調査に用いたベンチマーク関数を以下に示す。

$$f_1(x) = \sum_{i=1}^D (x_i)^2 \quad (3)$$

$$f_2(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (4)$$

$$f_3(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (5)$$

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (6)$$

f_1, f_2 は単峰性関数であり、 f_3, f_4 は多峰性関数である。

パラメータ設定はハチの数を 100、関数の呼び出し回数を 100,000、次元数を 30 とした。変更する変数の数を 2 から 30 まで増加させ、各 30 回実行した結果を図 1 に示す。

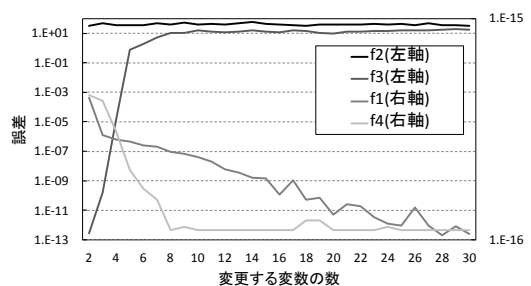


図 1 各ベンチマーク関数に対する探索性能

[†] 宇部工業高等専門学校, Ube National College of Technology

f_1 で最も良い最適値を示したのは28、 f_2 は18、 f_3 は2、 f_4 は30となった。次にABC、GABCによる探索を行い、性能を比較した。4つの関数の最適化の様子を図2-5に示す。縦軸は最適解との誤差、横軸は関数を呼び出した回数である。

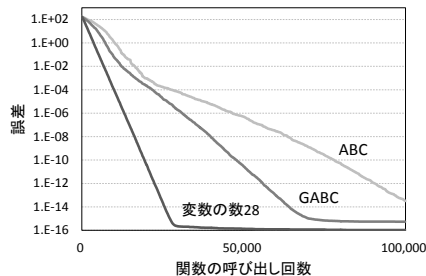


図2 ABC、GABC との比較 (f_1)

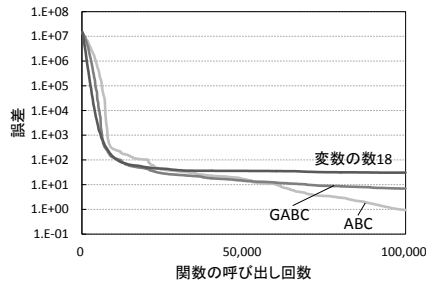


図3 ABC、GABC との比較 (f_2)

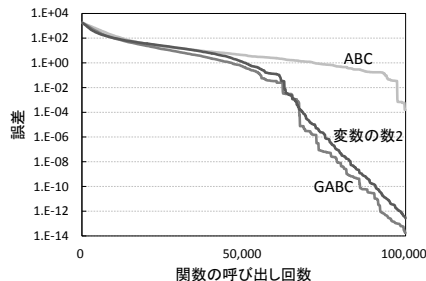


図4 ABC、GABC との比較 (f_3)

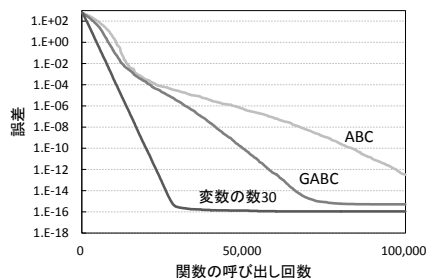


図5 ABC、GABC との比較 (f_4)

f_1 、 f_4 については変更する変数の数を増やすことでABC、GABCよりも良い性能を示した。さらにGABCと比べ、約2倍収束が速いことがわかる。 f_2 、 f_3 は序盤では両手法より小さい誤差だが、その後逆転されている。このことから f_2 、

f_3 は最適解近くでの複数の変数の変更は逆に探索性能を悪化させていることがわかる。

そこで、序盤に複数の変数を変更し、徐々に変更する変数を減らすアルゴリズムを提案する。 f_2 では変更する変数の数を30から関数を10回呼び出すごとに1つつ減らすと高い性能を示し、 f_3 では4から関数を10回呼び出すごとに1つつ減らすと高い性能を示した。図2、3と比較した結果を図6、7に示す。

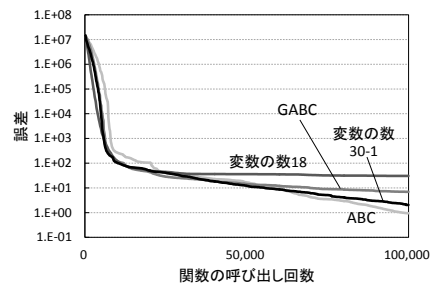


図6 提案手法の性能 (f_2)

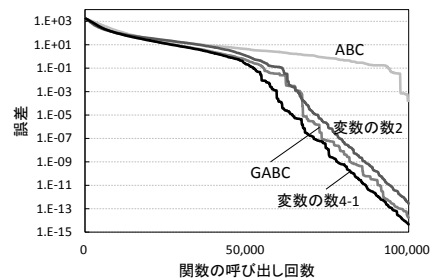


図7 提案手法の性能 (f_3)

f_2 ではABCに劣るもののGABCよりも高い性能を示した。 f_3 ではABCに対して約5桁、GABCに対して約1桁性能向上した。

4. おわりに

今回、GABCの変更する変数の数を増やすことで探索にどのような影響を与えるかを調査した。その結果、関数ごとに適した数の変数を変更することで収束速度と探索性能が向上することが分かった。

今後の展開としては、対象となるベンチマーク関数を増やす、変更する変数の数を変えた場合の各ハチ（変数 x ）の変化を調べることで探索の特徴を見つけさらなる探索性能向上を目指す、などが挙げられる。

参考文献

- [1] D. Karaboga, "AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION", Technical Report TR06, Erciyes University, Engineering Faculty, Computer 2005.
- [2] G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization.", Applied Mathematics and Computation, 3166-3173(2010).