高次元関数最適化における Artificial Bee Colony アルゴリズムの探索性能評価*

飯村伊智郎[†]・中山 茂[‡]

Search Performance Evaluation of Artificial Bee Colony Algorithm on High-Dimensional Function Optimization*

Ichiro IIMURA[†] and Shigeru NAKAYAMA[‡]

1. はじめに

ミツバチの群れによる知的な採餌行動に着想を得た Artificial Bee Colony (ABC) アルゴリズム [1,2] は, V. Tereshko らのモデル [3] に基づいて, 多変数関数や多峰性関数の最適化のために, D. Karaboga によって 2005 年に提案された [4]. その後, D. Karaboga らは, 進化的アルゴリズムの一種である遺伝的アルゴリズム (Genetic Algorithm: GA) や Differential Evolution (DE) アルゴリズム, そして群知能の一種である Particle Swarm Optimization (PSO) アルゴリズムを比較対象として, 関数最適化に関する計算機実験を行い, 制御パラメータ数がきわめて少ないという特徴を有する ABC は, GA や DE, PSO と同等以上の探索性能を有することを, 2009 年に参考文献 [5] で報告している. さらに, 関数の次元数が高くなるほど, ABC の有効性が顕著になると述べている. しかしながら, その次元数はたかだか 30 次元までであり, ABC の次元数に対するロバスト性を明らかにするためには, さらに高次元での評価が必要である. 本研究では, ABC と同じ群知能の一種である PSO を比較対象として, 低次元の先行研究と同様の実験条件のもとで, 30 次元以上の 100 次元までの関数を対象とした計算機実験を行い, ABC の次元数に対するロバスト性を評価した.

2. ABC アルゴリズムの概要

ABC におけるコロニー¹は, 三種類のミツバチ・グループから成り, それぞれの役割の概要は次の通りである.

Employed Bees (雇用ハチ): 各雇用ハチは, 特定の一つの食物源を担当し, その近傍を記憶し探索する. なお, 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になる.

Onlooker Bees (見物ハチ): 各見物ハチは, 巣箱内に戻ってきた雇用ハチのダンスの様子から食物源の位置や豊かさの情報を取得し, 食物源の一つを選びその近傍を探索する.

Scout Bees (偵察ハチ): 各偵察ハチは, ランダムに新たな食物源を探索する.

ABC では, 食物源の位置は対象問題の解候補を, 食物源の密度はその解候補の質, つまり適応度を表す. なお, 制御パラメータは, 食物源数である N_S , 雇用ハチが担当する食物源の探索をあきらめる断念反復回数 $limit$, そして探索を終了する最大反復回数 N_{MC} の三つのみであり, ほかは乱数を用いる. また, 食物源の数は雇用ハチ数と等しく, 雇用ハチと見物ハチの数は同数である.

以下に, ABC アルゴリズムを関数最適化問題へ適用し, 目的関数に対する最適解を探索する方法を述べる.

Step 1 [初期化] 目的関数の解候補 $x_i = [x_{ij}]$ をランダムに, $trial_i$ を 0 に初期化し, 適応度 $fit(x_i)$ を算出する処理を, すべての解候補に対して行う. ただし, $i \in \{1, 2, \dots, N_S\}$, $j \in \{1, 2, \dots, D\}$ であり, N_S は食物源数, D は目的関数の次元数を表し, $trial_i$ は解候補 x_i が雇用ハチや見物ハチによって更新されなかった回数である. なお, 解候補 x_i のランダムな初期化は (1) 式に従う.

$$x_{ij} = l_j + \text{rand}[0, 1] \cdot (u_j - l_j) \quad (1)$$

ただし, l_j と u_j は x_{ij} の下限値と上限値である.

Step 2 [雇用ハチによる探索] 各雇用ハチは, 担当する特定の一つの解候補 x_i に基づき, 新たな解候補 $v_i = [v_{ij}]$ を生成して, 適応度 $fit(v_i)$ を算出する. 新たな解候補 v_i の生成は, (2) 式で定義する.

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

ただし, $k \in \{1, 2, \dots, N_S\}$ であり, k はランダムに決定され i とは異なる値をとる. つぎに, $fit(x_i)$ と $fit(v_i)$ と

* 原稿受付 2010 年 11 月 29 日

[†] 熊本県立大学 総合管理学部 Faculty of Administration, Prefectural Univ. of Kumamoto; 3-1-100 Tsukide, Kumamoto city, Kumamoto 862-8502, JAPAN

[‡] 鹿児島大学 大学院 理工学研究科 Graduate School of Science and Engineering, Kagoshima Univ.; 1-21-40 Korimoto, Kagoshima city, Kagoshima 890-0065, JAPAN

Key Words: artificial bee colony, swarm intelligence, metaheuristics, function optimization.

¹ここでは, 集合して生活するミツバチの一群を指す.

を比較し, $\text{fit}(x_i) < \text{fit}(v_i)$ の場合, x_i を v_i で更新して $\text{trial}_i = 0$ とし, そうでない場合には trial_i を 1 だけ増やす. なお, この更新処理を greedy selection とよぶ.

これらの処理をすべての雇用ハチに対して行った後, (3) 式で定義される選択確率 p_i を求める.

$$p_i = \text{fit}(x_i) / \sum_{n=1}^{N_S} \text{fit}(x_n) \quad (3)$$

Step 3 [見物ハチによる探索] 各見物ハチは, 選択確率 p_i に基づいたルーレット選択で解候補 x_i を選択する. つぎに, その選択した解候補 x_i に基づき, 新たな解候補 v_i を (2) 式に従って生成し, 適応度 $\text{fit}(v_i)$ を算出する. その後, 雇用ハチ同様, greedy selection を行う.

これらの処理をすべての見物ハチに対して行う.

Step 4 [偵察ハチによる探索] もし, $\max\{\text{trial}_i\}$ が断念反復回数 limit 以上であれば, 解候補 x_a ($a = \arg\max_i\{\text{trial}_i\}$) に基づく探索はこの時点で断念する. 解候補 x_a を担当していた雇用ハチは偵察ハチとなり, x_a にかわる新たな解候補 x_i を (1) 式に従って決定する. その後, $\text{fit}(x_i)$ を算出し, $\text{trial}_i = 0$ とする.

Step 5 [終了判定] これまでに探索した最良解を記録する. 本アルゴリズムの反復回数が最大反復回数 N_{MC} に達していれば探索を終了し, でなければ Step 2 へ戻る.

3. 高次元関数最適化に関する評価実験

3.1 実験方法

本評価実験では, ABC と同じ群知能の一種である PSO を比較対象として, D. Karaboga らの低次元における先行研究を拡張し, 最適化の対象である関数の次元数 D を 30 次元から 100 次元まで 10 次元間隔で変化させ, ABC の次元数に対するロバスト性を評価する. 評価実験で対象としたベンチマーク関数は, D. Karaboga らの先行研究でも用いられた Table 1 に示す関数であり, これらを対象とした最小化問題を取り扱う. いずれの関数においても, その最小値は 0 である. これら四つの関数は, 単峰性 (Unimodal) や多峰性 (Multimodal), 分離可能 (Separable) や分離不可能 (Non-Separable)¹ など, それぞれの特徴が重ならないように選択した.

実験の条件は, D. Karaboga らの先行研究と同様であり, 両アルゴリズムに共通のパラメータである集団サイズ²は 50 とし, 最大評価回数を 500,000 とした. 集団サイズが 50 で, 最大評価回数が 500,000 であるため, アルゴリズムの最大反復回数 N_{MC} は, PSO, ABC とともに 10,000 となる. また, PSO に特有の粒子速度の更新式に関する五つのパラメータ値は, D. Karaboga らの先行研究と同じく参考文献 [6] にない, $(\omega, \phi_1, \phi_2) = (0.6, 1.8, 1.8)$,

また v_{max} は 0 から探索領域 (定義域) の下限と上限までの距離のうち長い方の距離の 15% とし, $v_{min} = -v_{max}$ とした³. 一方, ABC 特有のパラメータは断念反復回数 limit のみとなるが, D. Karaboga ら先行研究では $\text{limit} = N_S \times D$ としているため, 食物源数 N_S と対象とする関数の次元数 D から limit は自動的に決定され, ここでは実質的にパラメータはなくなる. 実験は一条件あたり 30 回試行し, 関数値の評価にはその平均値を用いるものとした. なお, 各試行における関数値が 10^{-12} 以下である場合, D. Karaboga ら先行研究と合わせて 0 と見なすものとした.

3.2 実験結果および考察

実験結果を Table 2 および Fig. 1 に示す. PSO の結果である Table 2(a) や Fig. 1(a) を見ると, 単峰性であるが設計変数間に依存関係を持つ Rosenbrock や設計変数間に依存関係はないが格子状に複数の準最適解を持つ多峰性である Rastrigin に関して, 次元数 D が大きくなるにつれて探索性能が著しく悪化していることがわかる. ところで Griewank については, 厳密には, 局所的には多峰性で依存関係が強いが, 大域的に見ると単峰性で依存関係が弱い関数であり, 大域的には Sphere と近い性質を有していると捉えられる. つまり PSO は, 設計変数間の依存関係がある場合や多峰性の場合には次元数増加に伴い探索性能の低下が見られるが, Sphere のように単峰性で設計変数間に依存関係がない場合や Griewank のように大域的には単峰性で依存関係が弱いような性質を有する関数に対しては, 次元数増加に伴う探索性能の低下は見られないと考えられる. 一方, ABC の結果である Table 2(b) や Fig. 1(b) によると, Rosenbrock 以外の関数について, 次元数 D が増加しても最適解 (最小値) を探索できていることが認められ, 単峰性であるが設計変数間に依存関係を持つ Rosenbrock に関しては PSO に比べて探索性能の著しい悪化は見られず, 探索された解の質も比較的良好であることが確認できる. つまり ABC は, 設計変数間の依存関係がある場合にのみ, 次元数の増加に伴い, PSO に比べれば良好な探索結果ではあるものの多少の探索性能低下が見られるが, 大域的に設計変数間に依存関係がないあるいは弱いような性質を有する関数に対しては, 単峰性や多峰性を問わず, 次元数増加に伴う探索性能の低下は見られないと考えられる.

このような結果となった理由としては, PSO の探索過程は, 過去の探索履歴 (パーソナルベストやグローバルベスト) に現在の探索が依存するが, ABC ではこれまでの探索場所を捨てて偵察ハチが過去の探索履歴に縛られることなくランダムに新たな探索場所を探す機構があることで, 多峰性関数に対しては局所解からの脱出を, 設計変数間に依存関係がある関数に対しては, より大域

¹ 分離可能 (Separable) とは設計変数間に依存関係がないことを, 分離不可能 (Non-Separable) とは設計変数間に依存関係があることを意味する.

² 集団サイズとは, PSO では粒子数を, ABC では雇用ハチ数と見物ハチ数の合計を指す.

³ PSO 特有のパラメータである $\omega, \phi_1, \phi_2, v_{min}, v_{max}$ の詳細は, 参考文献 [5] を参照していただきたい.

Table 1 Benchmark functions used in evaluation experiment (C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-separable)

No	Function	C	Range	Dimension	Optimum	Formulation
1	Sphere	US	$[-100, 100]$	$\{30, 40, 50, \dots, 100\}$	$f_1(0, 0, 0, \dots, 0) = 0$	$f_1(\mathbf{x}) = \sum_{i=1}^n x_i^2$
2	Rosenbrock	UN	$[-30, 30]$	$\{30, 40, 50, \dots, 100\}$	$f_2(1, 1, 1, \dots, 1) = 0$	$f_2(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
3	Rastrigin	MS	$[-5.12, 5.12]$	$\{30, 40, 50, \dots, 100\}$	$f_3(0, 0, 0, \dots, 0) = 0$	$f_3(\mathbf{x}) = \sum_{i=1}^n \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$
4	Griewank	MN	$[-600, 600]$	$\{30, 40, 50, \dots, 100\}$	$f_4(0, 0, 0, \dots, 0) = 0$	$f_4(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

Table 2 Experimental results of high-dimensional function optimization

(a) PSO algorithm

Function	C	Dimensions							
		30	40	50	60	70	80	90	100
Sphere	US	0	0	0	0	0	0	0	0
Rosenbrock	UN	5.5232662	22.1272991	33.6306826	59.5093669	74.6426053	99.0612379	114.5917377	148.4258244
Rastrigin	MS	29.3512678	43.2806763	50.5107067	60.8914444	68.3867981	76.8107740	90.0768810	104.6695947
Griewank	MN	0	0	0	0	0	0	0	0

(b) ABC algorithm

Function	C	Dimensions							
		30	40	50	60	70	80	90	100
Sphere	US	0	0	0	0	0	0	0	0
Rosenbrock	UN	0.0716225	0.1413067	0.1343018	0.1675062	0.3293138	0.3570076	0.3407050	0.4693217
Rastrigin	MS	0	0	0	0	0	0	0	0
Griewank	MN	0	0	0	0	0	0	0	0

(a) PSO algorithm

(b) ABC algorithm

Fig. 1 Dimensions vs. search performance

的な探索を実現できているのではないかと考える。

4. おわりに

評価実験の結果から、制御パラメータ数がきわめて少ないという特徴を有するABCは、同じ群知能の一種であるPSOと比較して、最適化の対象となる関数の特徴

に大きく左右されることなく、次元数の増加に対してロバストであることを示した。今後は、次元数の増加に対するロバスト性がABCアルゴリズムの何に起因するものなのかを明らかにすべく、より多くのベンチマーク関数を対象として評価実験を行う予定である。

謝 辞

本研究の一部は、(株)アリテックからの受託研究に関するものである。

参 考 文 献

- [1] X.-S. Yang: *Nature-Inspired Metaheuristic Algorithms*, Luniver Press (2008)
- [2] E.-G. Talbi: *Metaheuristics: From Design to Implementation*, Wiley (2009)
- [3] V. Tereshko and A. Loengarov: Collective decision-making in honeybee foraging dynamics; *Comput. Inf. Sys. J.*, Vol. 9, No. 3, pp. 1–7 (2005)
- [4] D. Karaboga: An idea based on honeybee swarm for numerical optimization; *Technical Report TR06*, Erziyes University, Engineering Faculty, Computer Engineering Department (2005)
- [5] D. Karaboga and B. Akay: A comparative study of artificial bee colony algorithm; *Applied Mathematics and Computation*, Vol. 214, pp. 108–132 (2009)
- [6] J. Vesterstrom and R. Thomsen: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems; *Proc. of the IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 1980–1987 (2004)