

ミツバチコロニーアルゴリズムにおける情報活用と探索の混同とその機能

A Confusion of Exploration-Exploitation in Artificial Bee Colony and its a Mechanism

小澤 優太*¹
Ozawa Yuuta

坂本 佑樹*¹
Sakamoto Yuki

大用 庫智*²
Kuratomo Oyo

高橋 達二*¹
Tatsuji Takahashi

*¹ 東京電機大学理工学部
School of Science and Technology, Tokyo Denki University

*² 東京電機大学大学院
Graduate School of Tokyo Denki University

人工蜂コロニー(ABC)法は採餌行動における分業を表現し、制約付き最適化問題における情報の活用と探索を両立する。解が確率的・ノイズな場合、偵察バチの配置は多本腕バンディット問題的形式を取るが、現実的には偵察バチの帰還時間と帰還自体が不定となり、従来の解法は不向きである。本研究では、粘菌・神経細胞・人間認知に見られる適応的特性である情報の活用と探索の混同を組み込み、ABC の性能向上を試みる。

1. はじめに

蟻や蜂といった社会性を持つ昆虫は、日々コロニーを作り生活している。蟻を観察すると、コロニーを構成している 1 匹 1 匹が知能を持っており、群れ全体が高度に組織化されているように見える。要するに社会性を持つ昆虫のコロニーでは、状況に応じ、個々の相互作用から集団行動が生まれている[Pater 10]。例えば蟻の場合、1 匹 1 匹は他の蟻が残したフェロモンを辿るだけだが全体から見ると、最短経路を探索するという難しい問題を解いている[亀田 07]。これは群知能と呼ばれているものである。最近ではアリのコロニーの原理を応用したアルゴリズムを使用し、生産システムや物流ルートを最適化されている試みがある[Pater 10]。

本稿では蜜蜂の動きを模した最適化アルゴリズムである Artificial Bee Colony (以下 ABC) を扱う。近年、昆虫の動きを模した最適化アルゴリズムの開発が活発に行われており、その中でも蜜蜂の採蜜行動は高い注目を集めている[Karl 05]。蜂の採蜜行動は、8 の字ダンスなどの特徴的な行動があることが解明されており[Karl 05]、その特徴を模した最適化アルゴリズムが各種提案されている[Karaboga 05; 古川 12]。最適化アルゴリズムの中でも ABC は蜜蜂の 8 の字ダンスを利用した行き先選択を、解の集約により表したものである。ABC の長所は単純な命令を組み合わせただけで非常に高い効率を得られること、また局所解から脱出しやすいという点である。しかし、探索する範囲がある程度広範囲に及ぶため、収束しにくいという問題点がある。

そこで我々は蜂のばらつきを嫌う特性の利用を提案する。例えば蜂は $6\mu\text{L}$ 、 $0\mu\text{L}$ 、 $0\mu\text{L}$ 蜜を含んだ 3 本の花からなる蜜源より、 $2\mu\text{L}$ ずつ一定量の蜜を含んだ 3 本の花からなる蜜源のほうを好むのである[菊沢 92]。これは一種の相対評価ではないかと考えられる。どちらもとれる蜜の量は一緒であるがまったく蜜が取れない花がいることにより、他の蜜源の評価が上がるのである。

もともとの ABC では解や蜜源の相対評価はされていない。そこで本稿では、相対評価を行うことができる Loosely Symmetric Model (以下 LS) [篠原 07] を ABC に適用し、その挙動を観察する。

連絡先: 小澤優太 email: 10rd050@ms.dendai.ac.jp

2. Artificial Bee Colony Algorithm

ABC は蜜蜂の採取行動を模した最適化手法である。ABC における主な構成要因は 3 つの動きに分けることができる。また、ここで言う調整とは解の試行という意味で、上書きはされない。更新とは解を試行した上で、上書きもされる。

- 収穫蜂: すべての蜜源の解を 1 つ調整、及び更新
- 追従蜂: 蜜源の適合度によって蜜源を選択し、解の調整、及び更新
- 偵察蜂: 一定期間更新されなかった蜜源をランダムな解と置き換える

ABC はこの 3 つの行動をまとめて 1 回とし、N セット繰り返すことにより最適化問題を解く。ABC の蜜源の初期設定は 10 個持っており、この蜜源は最適化問題における解の集合である。例としてある最適化問題の解が 5 つあるとする。この 5 つの解を ABC では 1 つの蜜源(まとまり)、として見る。ABC は蜜源を 10 個持っているので、単純計算 50 個の解を持っていることになる。この 50 個の解を調整することにより、最適化問題を解いていく。

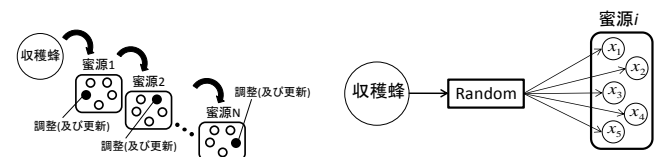


図 1: 採穫蜂における解選択図

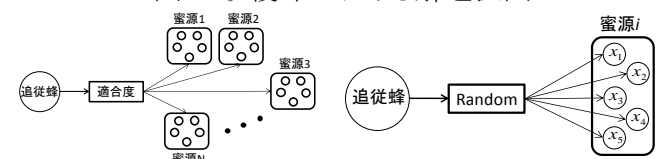


図 2: 追従蜂における蜜源及び解選択図

2.1 収穫蜂

この収穫蜂は蜜源(例だと 5 つ)全てを通過し、中身である解をランダムで 1 つずつ選び調整する(図 1)。もし調整した解が最適解と比較して良くなれば解を更新、最適解と比較して悪くなったら解は更新されない。更新に使用される式は(1)である。

$$v_{mi} = x_{mi} + \varphi_{mi}(x_{mi} - x_{ki}) \quad (1)$$

v_{mi} : 蜜源 m の調整された解、 x_{mi} : 蜜源 m の調整前の解、 φ_{mi} : -1~1 の乱数、 x_{ki} : ランダムで選ばれた蜜源 k の解

解の更新は v_{mi} と x_{mi} を比較し、もし v_{mi} が x_{mi} より良かったら v_{mi} で更新する。この収穫蜂は蜜源と同じ数だけおり、1匹の収穫蜂はすべての蜜源を訪れ、蜜源の中の解をランダムで選び調整を行う。

2.2 追従蜂

この追従蜂は蜜源を適合度によって選ぶ(図 2)。蜜源ごとに適合度を持っている。この適合度は蜜源の持つ解(例では 5 つ)が最適解とどれだけ合っているかを示す値である。適合度と言うのは解 1 つ 1 つではなく、解の集合で計算される。例えば 1 つだけとても良い解を持っているだけの蜜源の適合度はあまり高くない。適合度を求める式は(2)である。

$$fit_m(\vec{x}_m) = \begin{cases} \frac{1}{1 + f_m(\vec{x}_m)} & \text{if } f_m(\vec{x}_m) \geq 0 \\ 1 + abs(f_m(\vec{x}_m)) & \text{if } f_m(\vec{x}_m) < 0 \end{cases} \quad (2).$$

$fit_m: \vec{x}_m$ の適合度、 f_m : 目的関数、 \vec{x}_m : 解のまとまり

適合度の高い蜜源はより多くの追従蜂が訪れ、解の調整が行われる。この調整作業は収穫蜂と同様のものである。調整し、解がよくなれば適合度もその度更新される。逆に適合度の低い蜜源は追従蜂があまり訪れず、解の調整があまり行われない。追従蜂は蜜源の数だけ解の調整作業を行う。例えば、10 回解の調整作業が行なわれる蜜源もあれば、まったく解の調整されない蜜源も現れる。

2.3 偵察蜂

この偵察蜂は一定回数更新作業が行われなかった蜜源を新たな蜜源に置換する。収穫、追従蜂の調整作業を行った時、もし長時間解が更新されなかったら、その蜜源における解をすべてランダムに初期化する(ランダムの値は 5 章の関数を参照)。

ABC は収穫蜂、追従蜂、偵察蜂の流れを N 回繰り返して、最適解を求めていく。

3. 改良に用いたモデル

ABC を改良するために LS を使用した。LS は人の非論理的なバイアスを記述する対称性と相互排他性バイアスを緩く満たすモデルであり以下の表から次の式として定義される。ABC に適用するにあたり、一般化を施した[Kohno 12]。

表 1. $n \times 2$ の分割表と共変動情報

		判断の回数		
	外れ値の回数	x_i の選択回数		合計値
x_1	a_1	b_1		$n_1(a_1 + b_1)$
x_2	a_2	b_2		$n_2(a_2 + b_2)$
\vdots	\vdots	\vdots		\vdots
x_i	a_i	b_i		$n_i(a_i + b_i)$

$$LS(\text{外れ値}| \text{解の要素 } i) = \frac{a_i + \frac{b_H b_L}{b_H + b_L}}{a_i + b_i + \frac{a_H a_L}{a_H + a_L} + \frac{a_H a_L}{a_H + a_L}} \quad (3).$$

ここで、 a_H, b_H は最大の n_i を持つ a_i, b_i 、 a_L, b_L は最小の n_i を持つ a_i, b_i で、 a_i は外れ値であった回数、 b_i は x_i の選択回数である(詳細は 4.2 参照)。

人が推論する際に、 $p \rightarrow q$ が真ならば $q \rightarrow p$ も真である、という非論理的な思考が実際に存在する。これは先ほど述べた対称

性バイアスである。またその裏である $\bar{p} \rightarrow \bar{q}$ も真であると考えてしまう傾向を相互排他性バイアスという[篠原 07]。

3.1 対称性推論と相対評価

LS は対称性推論という人の認知的性質を参考にして提案された[篠原 07]。対称性理論とは人が考えるにあたって、 $p \rightarrow q$ が真ならばその逆、理論的には正しくない $q \rightarrow p$ もまた真であると渡された条件文を双方向的に考えてしまう人の思考特性である。これは論理的には正しくないのだが、人の身の回りに置くる要因としては双方向的に考えたほうが正しい場合が多い[Takahashi 10]。さらに言うならば裏命題である $\bar{p} \rightarrow \bar{q}$ も正しいと考えてしまう傾向があると知られている。これは対称性推論の推論形式に含まれ、相互排他性と呼ばれる。例えば、人が 2 人いて片方が失敗したならば、もう片方の人に期待(評価)が集まるなどが挙げられる。

具体例のようにまったく関連性のない、また関連があるかどうか分からない要因で他方への評価が上がるということは論理学においては正しくない。だが、現実世界の人は日ごろからこの評価方式を行ってしまう。このような評価方式は相対評価と呼ばれ、問題を解決するに当たって、成功したならばその手法に固執し、失敗したら他の方法をテストするような作用を發揮させる。要するに対称性推論に関係がないと一概には言えない事柄に関係を持たせることで、決定論的な価値ではなく、相対的な評価を下す。LS が多本腕バンディット問題等で優れた評価を出せるのはこの対称性推論により相対評価を実行しているからである[篠原 07]。

4. 実装方法

4.1 ABC の問題点

ABC の問題点として、ランダム要素が強いという点があげられる(図 1)。これは局所解に陥らないようにするための対策であると考えられるが、同時に問題点を抱えている。その問題点とは適合度に応じて蜜源に追従蜂をたくさん呼んだとしても、場合によってはほとんど調整されない解がでてしまうということである。調整されない解が出てしまう理由として、調整時、蜜源の中身である解はランダムで選択される点があげられる。適合度は蜜源全体で計算されるので、結果的に他の解が良かろうと 1 つの解が足を引っ張り適合度が適切に上昇しない可能性がでてしまう。これがどの蜜源でも続くと、一定以上解が改善されずアルゴリズムが終了してしまう。これが平均を計算したときの外れ値となり、結果が悪くなり、結果の振幅も大きくなってしまふ。

4.2 解決法

問題を解決する方法として、LS を収穫蜂の解の選択に適用した(図 3)。

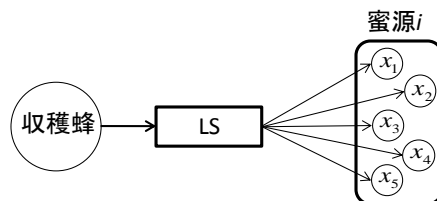


図 3: Artificial Bee Colony + LS

LS を用いた理由として蜜源の中身である解の外れ値を無くすることが目的である。外れ値は常に流動するため、LS の持つ緩い対称性推論が適切であると考えられる。蜜源全体の解が適切に

更新されれば、おのずと適合度も上昇し、解の調整される機会は増加されていく。本稿では外れ値を図4のように設定した。

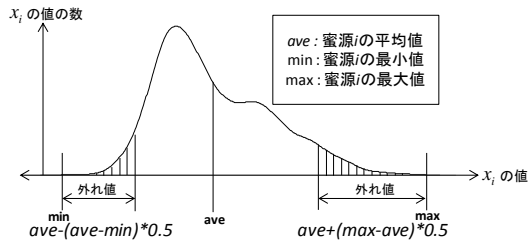


図4: 外れ値の設定

この外れ値を減少させるために用いるのが LS である。LS の重要な特徴は解の選択を柔軟に変更できる点であり、1 つの解の値が他の解の値に影響を与える点である。この作用により、解を局所解に陥らせることなく外れ値を減少させる。ABC に LS を実装する場合、収穫蜂が蜜源を選択した後、どの解を更新するか、選択する時に使用する(図3)。

また LS を用いた時の概要を図5に示す。

```

While
  ● 収穫蜂:LS の値を用いて解を選択し調整、及び更新
  ● 追従蜂:適合度で蜜源を選択し解を調整、及び更新
  ● 偵察蜂:一定期間更新されなかった解をランダムな解と置き換える
End
    
```

図5: ABC+LS アルゴリズムの概要

ABC において調整する解の決定はランダムで行われている。しかし、このままでは外れ値がでてきてしまう。これを解決するため解1つ1つに値を持たせた。値を持たせた理由として、どこが外れ値か認識する必要があったからである。値を出す式として(3)を使用した。値を上げるタイミングとしては、解の調整時、更新された、されなかったに関わらず、設定された外れ値に解が該当するならば値を上げる。例えば x_i を調整した時にもし、 x_i が外れ値に該当するならば a_i を1ずつプラスする。また同時に x_i を選択したことになるので、 b_i も1ずつプラスする。 a_i 、 b_i は初期化の時点で1を入れる。これは LS の式(3)において0乗算を避けるためである。外れ値時に値を上げる理由として、この ABC アルゴリズムは解1つ1つを見ているのではなく、解をある程度のみとまりで見ているからである。ABC に元から搭載されていた適合度は蜜源の評価関数である。

今回収穫蜂にしか LS を適用しなかったのは、この適合度を正確に上昇させるためである。適合度というのは追従蜂をより呼び込むためのものである。追従蜂は適合度で蜜源を選択し、解を LS ではなくランダムで選択することで解が偏らなくなり、局所解を防ぐ効果を生むためである。

$$v_{mi} = x_{mi} + \varphi_{mi}(x_{mi} - x_{ki}) \quad (4)$$

$$v_{mi} = x_{mi} + \varphi_{mi}(x_{mi} - x_{kv})$$

v_{mi} : 蜜源 m の調整された解[i], x_{mi} : 蜜源 m の調整前の解[i], φ_{mi} : -1~1 の乱数, x_{ki} : ランダムで選ばれた蜜源 k の解[i], x_{kv} : ランダムで選ばれた蜜源 k の解[v]

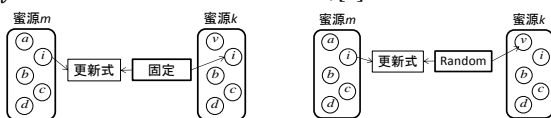


図6: 更新式のアルゴリズム

また、LS 用いる際の解の更新式を(1)の式から(4)のように変更した。変更した理由としては、調整に用いる際の i を収穫蜂と同様にしてしまうと、蜜源分しか、乱数が生まれなため、解が更新されなくなってしまう。そこで、 i 以外の v を用いることで更新の局所化を防いだ(図6)。

より蜜源に追従蜂を呼び込むには1つ良い解があるだけではさほど呼び込めない。追従蜂がこないということは調整回数が少ないということである。調整回数が少なければ、例えば全体で一番良い解がその蜜源にあったとしても一定以上に更新されることは少ない。そこで LS を用いて補助を行うこととした。全体の解をよくしたいので、もしある解を調整した時、その解が外れ値ならそこはもっと蜜がでる(解がよくなる) = 値が上がるとしたのである。これにより更新されない解が出てくることを防止しようとした。この工程を入れることにより結果として適切に適合度が上昇することとなり、より効率的に解を求めることが出来、同時に結果のばらつきを抑えようと考えた。

5. シミュレーション

本稿では、4種類のベンチマーク関数で ABC に相対評価が有効かどうかテストする。ABC と ABC に LS を追加したもので比較をするのである。ABC のソースはホームページ [Karaboga] から引用した。また初期設定として 2500 回を 1 試行とし、100 回繰り返し平均をとった。蜜源の数は 10、蜜源の中身である解の個数を 30、50、100、200 (以下 D) と変化させた。以下にそれぞれ関数の数式とその最適解を示す。

(1) Sphere 関数

$$f(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (5)$$

問題の範囲は $[-100 < x < 100]$

最適解は $f(\vec{0})=0$

(2) Rosenbrock 関数

$$f(\vec{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + 100(x_i - 1)^2] \quad (6)$$

問題の範囲は $[-30 < x < 30]$

最適解は $f(\vec{1})=0$

(3) Griewank 関数

$$f(\vec{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (7)$$

問題の範囲は $[-600 < x < 600]$

最適解は $f(\vec{0})=0$

(4) Rastrigin 関数

$$f(\vec{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (8)$$

問題の範囲は $[-5.12 < x < 5.12]$

最適解は $f(\vec{0})=0$

5.2 シミュレーション結果

D を 30、50、100、200 と変化させた結果を表2から表8に示す。比較相手として ABC と ABC に LS を組み込んだもので比較した。今回は解の振れ幅の改善、また解自体の改善を目指しており、D を大きくすると振れ幅もまた大きくなるので大きめの値を試行した。AVG が平均で STD が標準偏差である。

まず sphere 関数に関してだが、表 2 が改変前の ABC で、表 3 が LS を組み込んだ ABC である。見比べると LS のほうが良い結果をだしていることが分かる。

次に Rosenbrock 関数だが、表 4 が改変前の ABC で、表 5 が LS を組み込んだ ABC である。見比べると LS のほうが良い結果をだしていることが分かる。

次に Griewank 関数だが、表 6 が改変前の ABC で、表 7 が LS を組み込んだ ABC である。D が少ない時に平均が多少芳しくないものであったが、標準偏差を小さくできたので振幅の縮小には成功している。

最後に Rastrigin 関数であるが、表 8 が改変前の ABC で、表 9 が LS を組み込んだ ABC である。見比べると LS のほうが良い結果をだしていることが分かる。

以上のことから当初の目標であった収束のしにくさの改善はある程度行えたのではないか。その理由としては解の更新が LS によって偏らず、適切に適合度が上昇したものと考えられる。また ABC に LS を加えることは効果的であると言える。

6. 考察と結語

本稿では、収穫蜂の解の選択に LS を用いることで ABC の信頼性、また、結果の改善を目指した。その結果、収穫蜂の解の選択に LS を用いたものは関数近似のシミュレーションで良い結果を得ることが出来た。これは蜂がばらつきを嫌う傾向を利用した結果であり、それゆえ局所解を回避し、より早く最適解に収束できたのである。これより、ABC に LS を加えることは有効であり、相対評価が良い影響を及ぼしたと言える。特に D が多い時、高次元の問題時に効果を発揮する。これは LS により蜜源の内部である解を緩く判断し、適切に解を更新した結果である。今後の目標は近似関数の問題だけではなく、様々な問題の検証を行うことである。

参考文献

- [篠原 07] 篠原修二, 田口亮, 桂田浩一, 新田恒雄. 因果性に基づく信念形成モデルと N 本腕バンディット問題への適用, 人工知能学会論文誌, Vol.22, No.1, pp.58-68, 2007.
- [Karaboga 05] Karaboga D. (2005). An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- [古川 12] 古川まき, 森敏彦, 鈴木泰博. 蜜蜂の蜜源選択における閾値調整機能を模した最適化アルゴリズム, 知能情報(日本知能情報ファジィ学会誌) Vol24, No6, pp.1090-1100(2012)ineering Department.
- [古川 11] 古川まき, 鈴木泰博. ミツバチの採餌行動を模した最短経路探索法 動的な環境での挙動, 人工知能学会, 3C1-OS14-8, 2011
- [Pham 06] Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., and M.Zaidi "The Bees Algorithm ? A Novel Tool for Complex Optimisation Problems", Proceedings of IPROMS 2006 Conference, pp.454-461, 2006
- [白崎 10] 白崎雄大, 下村将, 杉本雅樹, 上手洋子, 西尾芳文, 社団法人電子情報通信学会, NLP2010-113, 2010
- [菊沢 92] 菊沢喜八郎, 植物と花粉媒介者(ポリネーター), 光珠内季報, 年一巻号 p14-17, 1992
- [Peter 10] Peter Miller, 土方奈美[訳], 群れのルール, 東洋経済新報社, 2010
- [亀田 07] 亀田陽介, 狩野均, 局所最適解をフェロモン初期化に用いたアントコロニー最適化法による TSP の解法, 社団法人情報処理学会, 2007-MPS-63, 2007
- [Karl 05] Karl von Frisch, 伊藤智夫訳, ミツバチの不思議, 法政大学出版局, 2005X13
- [Kohno 12] Kohno, Y., Takahashi, T. (2012), "Loosely Symmetric Reasoning to Cope with The Speed-Accuracy Tradeoff", SCIS-ISIS 2012, Kobe Convention Center (KobePortopia Hotel), pp.1166-1171.

[Takahashi 10] T. Takahashi, M. Nakano, S. Shinohara (2010), "Cognitive symmetry: Illogical but rational biases", Symmetry: Culture and Science, Vol. 21, No. 1-3, 275-294.

表 2 通常 ABC:sphere 関数の平均と標準偏差

D	30	50	100	200
AVG	9.64E-16	4.54E-15	7.31E-06	0.112199
STD	2.95E-16	5.54E-15	1.32E-05	0.240052

表 3 ABC+LS:sphere 関数の平均と標準偏差

D	30	50	100	200
AVG	1.25E-15	2.42E-15	6.70E-09	1.82E-03
STD	4.91E-16	6.90E-16	6.59E-09	6.16E-04

表 4 通常 ABC:Rosenbrock 関数の平均と標準偏差

D	30	50	100	200
AVG	1.542078	3.44428	124.4362	1011.939
STD	2.650139	4.411457	85.17316	838.8149

表 5 ABC+LS:Rosenbrock 関数の平均と標準偏差

D	30	50	100	200
AVG	1.02E-01	3.45E-01	8.07E+00	3.82E+02
STD	1.74E-01	6.42E-01	2.30E+01	1.33E+02

表 6 通常 ABC:Griewank 関数の平均と標準偏差

D	30	50	100	200
AVG	2.61E-16	1.11E-15	2.03E-07	0.004299
STD	3.57E-16	1.77E-15	3.03E-07	0.015046

表 7 ABC+LS:Griewank 関数の平均と標準偏差

D	30	50	100	200
AVG	1.14E-15	2.72E-15	5.00E-11	1.29E-04
STD	3.47E-16	8.39E-16	5.29E-11	5.94E-05

表 8 通常 ABC:Rastrigin 関数の平均と標準偏差

D	30	50	100	200
AVG	0.042596	0.538423	15.40794	129.209
STD	0.196075	0.770988	4.418694	16.37304

表 9 ABC+LS:Rastrigin 関数の平均と標準偏差

D	30	50	100	200
AVG	4.73E-15	3.14E-09	3.17E-01	5.92E+01
STD	1.53E-14	3.11E-08	5.87E-01	8.50E+00