

## 第5章 蜂は8の字を書く：ABC アルゴリズム

アリと並んでハチも社会性昆虫として有名です（図 4.24）。ミツバチは以下の3つのタイプに分類されます。

- 働きバチ (employed bees)
- 傍観バチ (onlooker bees)
- 斥候バチ (scout bees)

働きバチは自分の記憶している餌場の近くを飛んで、餌についての情報を傍観バチに伝えます。傍観バチは働きバチからの情報を用いて餌場から最良の餌を選択的に探し出します。餌場についての情報が古くさくなると、働きバチはその情報を捨てて斥候バチになり新しい餌場を探しに行きます。ハチの巣の目的は最も効率の良い餌場を見つけることです。一般には巣の全体の半分が働きバチ、10–15% が斥候バチ、残りが傍観バチであるとされています。

この考え方に基づいて Karaboga は ABC アルゴリズム (Artificial Bee Colony) という最適化手法を提案しました。ABC アルゴリズムはミツバチの餌集めをまねた集団的探索法です。遺伝的アルゴリズムや PSO などに比べて少ない制御パラメータであることが、ABC の利点の1つとなっています。

ABC アルゴリズムでは、人工的なハチの群れは働きバチ、傍観バチ、斥候バチに分けられます。問題に対して  $N$  個の  $d$  次元の解が生成されて餌場

として参照されます。それぞれの働きバチは特定の餌場  $\mathbf{x}_i$  に割り当てられ、以下のオペレータを用いて新しい餌場  $\mathbf{v}_i$  を探します。

$$v_{ij} = x_{ij} + \text{rand}(-1, 1) \times (x_{ij} - x_{kj}), \quad (5.1)$$

ここで  $k \in \{1, 2, \dots, N\}$ ,  $k \neq i$  と  $j \in \{1, 2, \dots, d\}$  はランダムに選ばれたインデックスです。  $v_{ij}$  は  $\mathbf{v}_i$  ベクトルの  $j$  番目の要素です。つまり  $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id})^T$ ,  $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})^T$  となります。もしも新しい位置が定義域を外れているなら、許容値に戻されます。そして得られた  $\mathbf{v}_i$  は  $\mathbf{x}_i$  と比べられてよい餌場の方が選択されます。

働きバチと異なり、傍観バチは式 (5.1) を用いて餌場の空間をさらに探索し、より好ましい餌を選び出します。この選好スキームは働きバチからのフィードバック情報に基づいています。

では、ABC アルゴリズムの詳細を説明しましょう。

まず、餌場  $\mathbf{x}_i$  を探索する確率を以下のように計算します [11]。

$$p_i = \frac{fit_i}{\sum_{j=1}^N fit_j}, \quad (5.2)$$

$fit_i$  は  $i$  番目の餌場  $\mathbf{x}_i$  の適合度  $f(\mathbf{x}_i)$  です。簡単のため、適合度は非負であるとし、大きいほど良いとします。ハチはより良い餌場を好みます。もしも  $\mathbf{v}_i$  が  $\mathbf{x}_i$  よりも餌としての価値があれば、傍観バチは  $i$  番目の餌場  $\mathbf{x}_i$  を探している働きバチにその情報を伝え、記憶をリセットさせます。またある餌場が予め決められた繰り返し回数 (Limit) 内に更新されなければ (その近くでより良い餌場が見つからなければ)、この餌場は放棄されます。この餌場を用いていたハチは斥候となり新しい餌場に行かされます。標準的な ABC では、斥候バチはランダムな初期化により新しい餌を探します。

以上から ABC アルゴリズムは次のようになります。

**Step 0: 準備** 探索点の総数 ( $N$ ), 移動の総回数 ( $T_{max}$ ), 斥候の制御パラメータ (Limit) を設定する。働きバチと斥候バチの数は探索点の総数 ( $N$ )

に等しくする。目的関数  $f$  の値は非負であり、大きいほど良いものとする。

**Step 1: 初期化 1** 移動のカウンタ  $k$  を 1 とし、探索点更新数  $s_i$  を 0 とする。各探索点の初期位置ベクトル  $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})^T$  をランダムな値に設定する。ただし添え字  $i$  ( $i = 1, \dots, N$ ) は探索点のインデックスであり、 $d$  は探索空間の次元数である。

**Step 2: 初期化 2** 初期の最良解  $best$  を決定する。つまり以下のようにする。

$$i_g = \arg \max_i f(\mathbf{x}_i) \quad (5.3)$$

$$best = \mathbf{x}_{i_g} \quad (5.4)$$

**Step 3: 働き蜂の探索** 以下の式を用いて、現在の位置ベクトル  $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})^T$  から新しい位置ベクトル  $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id})^T$  を計算する。

$$v_{ij} = x_{ij} + \phi \cdot (x_{ij} - x_{kj}), \quad (5.5)$$

ここで、 $j$  はランダムに選ばれた次元番号、 $k$  は ( $i$  以外の) あるランダムに選ばれたインデックス、 $\phi$  は  $[-1, 1]$  の範囲で一様ランダムに生成された乱数である。位置ベクトル  $\mathbf{x}_i$  と探索点更新数  $s_i$  は以下の式に基づいて決定される。

$$I = \{i \mid f(\mathbf{x}_i) < f(\mathbf{v}_i)\} \quad (5.6)$$

$$\mathbf{x}_i = \begin{cases} \mathbf{v}_i & i \in I \\ \mathbf{x}_i & i \notin I \end{cases} \quad (5.7)$$

$$s_i = \begin{cases} 0 & i \in I \\ s_i + 1 & i \notin I \end{cases} \quad (5.8)$$

**Step 4: 傍観バチの探索** 以下の 2 つのステップを実行する。

## 1. 探索点の相対的なランク付け

各探索点の評価値に基づいて、相対確率  $P_i$  を適合度  $fit_i$  から計算する．  $fit_i = f(\mathbf{x}_i)$  であることに注意せよ．

$$P_i = \frac{fit_i}{\sum_{j=1}^N fit_j} \quad (5.9)$$

## 2. ルーレット選択と探索点の更新

上で計算した確率  $P_i$  に基づいて、更新すべき探索点を選び出す．探索点選ばれたあとで、Step 3にある手続を実行し、探索点の位置ベクトルを更新する．この選択と更新を  $N$  回まで繰り返す．

**Step 5: 斥候バチの探索**  $s_i \geq \text{Limit}$  であるような探索点があるとき、乱数を用いて生成された探索点と交換する．そしてその  $s_i$  を0にリセットする．

**Step 6: 最適解の更新** 最適解  $best$  を更新する．つまり、

$$i_g = \arg \max_i f(\mathbf{x}_i) \quad (5.10)$$

とする．このとき  $f(\mathbf{x}_{i_g}) > f(best)$  なら、

$$best = \mathbf{x}_{i_g} \quad (5.11)$$

とする．

**Step 7: 終了の決定** もし  $k = T_{max}$  ならば終了する．さもなければ、 $k = k + 1$  として Step 3に戻る．

ABC には多くの拡張の試みがなされています．たとえば、ABC はパラメータの外乱に弱いという欠点が指摘されています．この欠点を克服するため、ハチの餌探しの協調メカニズムに非分離のオペレータを導入した研究などがあります [9]．