

# **Artificial Bee Colony**

## **Tutorial**

für die Vorlesung

**Advanced Software Engineering**

des Studiengangs Angewandte Informatik  
an der Dualen Hochschule Baden-Württemberg Mosbach

von

**5703004, 1716504**

25. April 2022

# 1 Metapher

Der Artificial Bee Colony (ABC) Algorithmus ist vom Verhalten von Honigbienen inspiriert, die Nektar von Blumen sammeln und als Futter in den Bienenstock bringen. Dabei werden zunächst Kundschafterbienen (*scout bees*, im Folgenden *Scouts* genannt) ausgesendet, die nach Regionen suchen, in denen Nektar zu finden ist. Diese kehren schließlich zurück in den Bienenstock und informieren andere Bienen über die Fundstelle und Qualität (*fitness*) der Futterquelle. Daraufhin kehren die Scouts zur Futterstelle zurück. Einige andere Bienen folgen ihnen. Ein kleiner Teil der Scouts sucht daraufhin nach neuen Futterquellen, während andere Bienen weiterhin die Qualität einer Futterquelle bei jeder Rückkehr in den Bienenstock kommunizieren.

Dieses Verhalten sorgt für eine effiziente Nahrungssuche bei Honigbienen und wird für die iterative Suche nach einer Lösung von Optimierungsproblemen im ABC imitiert.

# 2 Strategie

Um ein globales Optimum zu finden, wird eine Honigbienenkolonie imitiert, die drei Arten von Bienen beinhaltet, die sog. *Employed Bees* (beschäftigten Bienen), die *Onlookers* (Zuschauerbienen) und die *Scouts* (Kundschafterbienen). Eine Hälfte der Kolonie besteht aus Employed Bees, die andere Hälfte beinhaltet die Onlookers und Scouts.

Futterquellen repräsentieren Lösungen eines Problems, das mit dem ABC gelöst werden soll. Je höher die Qualität bzw. Fitness einer Lösung ist, desto größer ist die Menge an Nektar. Der Algorithmus konvergiert, indem die Bienen iterativ das globale Optimum finden, da sie mit der Zeit die Futterquellen mit der besten Fitness ansteuern. Diese Futterquelle stellt die beste Lösung des Optimierungsproblems dar.

Scouts suchen zufällig nach einer neuen Futterquelle und ersetzen damit Futterquellen, also Lösungen, die sich nicht bewährt haben, deshalb von den Bienen verlassen wurden und damit nicht das globale Optimum darstellen.

Employed Bees suchen nach Futterquellen in ihrer direkten Nachbarschaft, die mehr Nektar haben, als die aktuelle Futterquelle, auf der sie sich befinden. Die Suche ist zwar zufällig, bezieht allerdings das globale Wissen des Schwarms über andere Futterquellen mit ein. Diese Art von Bienen sucht nach neuen Lösungen und teilt ihre Ergebnisse den Onlookers mit.

Onlookers wiederum beziehen die Positionen der Futterquellen von den Employed Bees. Diese Bienen wählen stetig bessere Lösungen aus und suchen in ihrer Nachbarschaft nach weiteren Lösungen.

Der ABC besteht aus drei Phasen. In der ersten Phase des Algorithmus werden verschiedene Positionen im Suchraum zufällig als Futterquellen ausgesucht und deren Fitness berechnet. In der zweiten Phase schwärmen die Employed Bees aus und suchen nach neuen Futterquellen in direkter Nachbarschaft zu den initialen Futterquellen. Wenn die Onlookers ausschwärmen, präferieren sie die Positionen, die am meisten Nektar produzieren. Diese Information wird ihnen durch die Employed Bees mitgeteilt. Wenn die Menge an Nektar ansteigt, steigt auch die Wahrscheinlichkeit, dass Onlookers diese Futterquelle auswählen. Diese Auswahl erfolgt in der dritten Phase des Algorithmus. Wird die Zielposition von einer Biene erreicht, so wird nach einer neuen Futterquelle in der Umgebung in der direkten Nachbarschaft gesucht. Wird eine Futterquelle verlassen, wird diese von den Scouts zufällig durch eine neue ersetzt.

So wird der gesamte Suchraum iterativ nach dem Optimum (Futterquelle mit der besten Fitness) abgesucht.

### 3 Prozedur

Zunächst werden zufällig Positionen im Suchraum ausgewählt. Die Anzahl der Positionen ist dabei von der Population abhängig. Jede Position (jede Futterquelle) ist ein  $n$ -dimensionaler Lösungsvektor des Optimierungsproblems, wobei  $n$  die Anzahl der Optimierungsparameter darstellt.

Nach der Initialisierung werden die einzelnen Positionen zyklisch von Employed Bees, Onlookers und Scouts besucht. Employed Bees aktualisieren dabei ihre aktuelle beste Position in Abhängigkeit der lokalen Information über die Fitness ihrer Futterquelle und testet die Menge an Nektar (Fitness) von neuen Quellen. Diese neuen Futterquellen sind mögliche neue Lösungen des Optimierungsproblems. Wenn die Nektarmenge der neuen Lösung höher ist als die momentane, wird die beste lokale Position einer Employed Bee aktualisiert. Ist die Nektarmenge kleiner als die der aktuellen Position, so merkt sich die Biene diese als beste lokale Position.

Nachdem alle Bienen ihren Suchvorgang abgeschlossen haben, teilen sie die neu gewonnen Informationen über die Menge an Nektar und die dazugehörigen Positionen den Onlookers mit. Die Onlookers berücksichtigen bei der Auswahl der Position, die sie anfliegen, die Menge an Nektarvorkommen aller gefundenen Futterquellen. Die Wahrscheinlichkeit, mit der eine Biene eine Futterquelle anfliegt, berechnet sich nach der folgenden Formel:

$$p_i = \frac{\text{fit}_i}{\sum_{j=1}^n \text{fit}_j} \quad (1)$$

wobei  $\text{fit}_i$  die Fitness bzw. die Menge an Nektar der Futterquelle einer Lösung  $i$  und  $n$  die Anzahl an Futterquellen ist. Diese Anzahl ist gleich der Anzahl an Employed Bees.

Um aus dem Gedächtnis einen neuen Kandidat für die Futterquelle zu erzeugen, kann die folgende Formel angewendet werden:

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

wobei  $k$  ein zufälliger Index der möglichen Lösungen und  $j$  ein zufälliger Index des Vektors an Optimierungsparametern ist.  $\Phi$  ist eine zufällige Zahl zwischen  $[-a; a]$ .

## 4 Pseudocode

## 5 Exploration and Exploitation

## 6 Entscheidungsregeln für Schwarmverhalten

## 7 Parameterabhängigkeiten

## 8 Zusammenspiel *Employed Bee*, *Onlooker Bee*, und *Scout Bee*