

# **Particle Swarm Algorithmus**

## **Tutorial**

für die Vorlesung

**Advanced Software Engineering**

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

von

**5703004, 1716504**

21. April 2022

# 1 Metapher

Der Particle Swarm Optimization (PSO) Algorithmus imitiert das Verhalten von Zugvögeln, die in einem Schwarm fliegen. Die theoretischen Grundlagen wurden dabei von Craig Reynolds bereits 1986 gelegt, der dieses Verhalten mit einem Programm namens *Boids* simuliert. Die Idee des Algorithmus besteht darin, dass einzelne Agenten (Partikel) miteinander interagieren, indem sie zunächst selbstständig den Suchraum erkunden und anschließend das neu erlangte Wissen über ein kollektives Gedächtnis miteinander teilen. Jeder Partikel hält dabei ein lokales Optimum ( $pBest$ ), zusammen mit den entsprechenden Positionskoordinaten in dem Suchraum und steuert dies dem kollektiven Gedächtnis bei, sodass der Schwarm iterativ ein globales Optimum ( $gBest$ ) ermitteln kann.

# 2 Strategie

Zunächst werden alle Agenten (Partikel) mit einer zufälligen Geschwindigkeit und Position initialisiert. Anschließend werden die jeweiligen Eigenschaften eines Partikels durch das Inertia, das kognitive individuelle Verhalten eines Partikels und das soziale Verhalten gegenüber aller Partikel aktualisiert. Ziel ist es, dass alle Agenten das Optimum in einem multidimensionalen Suchraum finden.

# 3 Prozedur

Der PSO nutzt eine vordefinierte Menge an Partikeln, die sich durch den Suchraum bewegen und dabei sowohl von ihrem letzten besten Punkt, dem letzten besten Punkt des Schwarms, als auch ihren nächsten Nachbarn beeinflusst werden. Dabei setzt sich nach jeder Iteration die Geschwindigkeit eines einzelnen Partikels aus der Summe der aktuellen Geschwindigkeit eines Partikels, dem individuellen kognitiven Verhalten eines Partikels und seinem sozialen Verhalten im gesamten Schwarm zusammen. Dabei werden die Parameter für das individuelle kognitive und soziale Verhalten mit festen, zu parametrisierenden, Koeffizienten gewichtet und jeweils mit einem zufälligen Wert zwischen 0 und 1 multipliziert. Anschließend werden die Ergebnisse jeweils mit der Differenz aus dem jeweiligen Optimum und der aktuellen Position des Partikels multipliziert. Daraus lässt sich die folgende Formel für die Aktualisierung der Geschwindigkeit je Iteration ableiten:

$$v_i(t + 1) = v_i(t) + (c_1 * \text{rand}() * (p_i^{\text{best}} - p_i(t))) + (c_2 * \text{rand}() * (p_{g\text{best}} - p_i(t))) \quad (1)$$

Anschließend wird die Position in jeder Iteration aus der aktuellen Position eines Partikels und der zuvor berechneten Geschwindigkeit berechnet:

$$p_i(t + 1) = p_i(t) + v_i(t) \quad (2)$$

## 4 Pseudocode

Der PSO benötigt zwei Eingabeparameter, die zu Beginn festgelegt werden. Das ist zum einen die Anzahl der Dimensionen (beim Travelling-Salesman-Problem beispielsweise die Anzahl der Orte), zum anderen die Anzahl der im Schwarm befindlichen Partikel. Dabei liegt die optimale Anzahl der Partikel in einem Bereich zwischen 10 und 30. Der Algorithmus liefert das Partikel, welches das globale Optimum gefunden hat.

Zu Beginn des Algorithmus wird die Population und das globale Optimum mit dem Standardwert 0 initialisiert. Anschließend erhalten entsprechend der festgelegten Größe der Population nacheinander die Partikel eine zufällige Geschwindigkeit und Position im Suchraum. Anschließend werden für das jeweilige Partikel die Kosten abhängig von ihrer Position berechnet. Die aktuelle Position entspricht zu Beginn dem lokalen Optimum ( $pBest$ ). Wenn die Kosten des lokalen Partikels kleiner als die bereits bekannten Kosten im kollektiven Gedächtnis sind, dann wird die globale beste Position mit der lokalen Position des Partikels überschrieben.

Anschließend wird die jeweilige Geschwindigkeit und Position der Partikel aktualisiert, die Kosten an der jeweiligen Position neu berechnet und ggf. als neues lokales Optimum gesetzt, bis die Abbruchbedingung erfüllt ist. Eine Abbruchbedingung kann beispielsweise das Erreichen einer maximalen Anzahl an Iterationen sein oder ein Fehlerwert unter einen vordefinierten Schwellwert fällt. Dann kann davon ausgegangen werden, dass ein globales Optimum gefunden wurde. Sind die Kosten der aktuellen Position eines Partikels kleiner als die des bisherigen globalen Optimums, wird auch dieser Wert angepasst.

## 5 Exploration and Exploitation

Bei der Exploration erkunden Partikel den noch unbekannten Suchraum nach einem neuen Optimum. Wie stark dieser Suchraum von den Partikeln erkundet wird, hängt maßgeblich von seinem kognitiven individuellen Verhalten ab, das durch die Konstante beeinflusst  $c_1$  wird.

Die Exploitation sorgt vor allem dafür, dass vielversprechende, bereits bekannte Regionen genauer von den Partikeln untersucht werden. Diese Regionen werden durch das kollektive Gedächtnis des Schwarms festgelegt. Wie stark ein Partikel exploriert, wird damit maßgeblich

durch die Konstante  $c_2$  und das harmonisierte Flugverhalten bez. der Position und Geschwindigkeit der Partikel (Intertia) beeinflusst.

Für den Erfolg des PSO ist eine ausgewogene Balance zwischen Exploration und Exploitation unabdingbar.

## **6 Entscheidungsregeln für Schwarmverhalten**

## **7 Parameterabhängigkeiten**

## **8 Zusammenspiel *gBest* und *pBest***