

1 Einleitung

„Das Ganze ist mehr als die Summe seiner Teile.“

ARISTOTELES

Man nehme eine zufällig verteilte Population von Agenten X . Es gilt $X = \{x_1, x_2, \dots, x_N\}$ mit $N \in \mathbb{N}$ als Anzahl der Agenten. Für einen Agenten $x_i \in X$ gilt

$$x_i = Lb + r \cdot (Ub - Lb) \quad \text{mit } i = 1, 2, \dots, N, \quad (1.1)$$

wobei Lb das untere und Ub das obere Limit des Definitionsbereichs ist.

Für die Nachbarschaft eines Agenten x_i wird der euklidische Abstand verwendet. Mithilfe eines Radiuses r wird die Nachbarschaft $\mathcal{N}_i \subseteq X$ eines Agenten x_i definiert als

$$\mathcal{N}_i = \{x \in X \mid d(x_i, x) \leq r\} \quad \text{mit } d(x_i, x) = |x_i - x|. \quad (1.2)$$

Zur Ermittlung der Nachbarschaft im n -dimensionalen Raum wird folgender Algorithmus verwendet:

Listing 1.1: Algorithmus zur Bestimmung der Nachbarschaft eines Agenten im n -dimensionalen Raum

```
1 def get_neighborhood(agent, all_agents, radius):
2     neighborhood = []
3     for other_agent in all_agents:
4         distance = euclidean_distance(agent.position, other_agent.position)
5         if distance <= radius:
6             neighborhood.append(other_agent)
7     return neighborhood
```

Füge Text
für Einlei-
tung hinzu