

UNIVERSITÄT LEIPZIG

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

INSTITUT FÜR INFORMATIK

# Simulation und Analyse von Strategien zur Eindämmung von Epidemien in Multi-Agenten-Systemen

Bachelorarbeit

Leipzig, 20. September 2023

*vorgelegt von:*

Philipp Riemer

pr83zyfu@studserv.uni-leipzig.de

Studiengang: Informatik B.Sc.

Matrikelnummer: 3725283

*Prüfer:*

Dr. Nicolas Wieseke

(Professur für Schwarmintelligenz und Komplexe Systeme)

*Betreuer:*

M. Sc. Tobias Jagla

(Professur für Schwarmintelligenz und Komplexe Systeme)

# Inhaltsverzeichnis

<b>Symbolverzeichnis</b>	<b>3</b>
<b>1 Einleitung</b>	<b>5</b>
1.1 Zielsetzung	5
1.2 Abgrenzung	6
1.3 Aufbau der Arbeit	6
<b>2 Grundlagen</b>	<b>8</b>
2.1 <i>SEIR</i> -Modell	8
2.2 Multi-Agenten-Systeme ( <i>MAS</i> )	9
2.3 <i>Correlated Random Walk (CRW)</i>	10
2.4 Pheromone	11
2.4.1 Biologischer Hintergrund	11
2.4.2 Modellierung von Pheromon-Simulationen	12
2.4.3 Anwendungsmodelle für <i>MAS</i>	13
2.5 Pheromon basiertes <i>Foraging</i>	13
2.6 DeziBot	14
2.6.1 Anwendung in der Forschung	14
2.7 Software-Simulation	14
<b>3 Methoden</b>	<b>16</b>
3.1 Allgemeines Modell	16
3.2 Spielfeldmodell	16
3.3 Ressourcenmodell	17
3.4 Pheromonmodell	18
3.4.1 Aufbau und Funktionsweise der Pheromone	18
3.4.2 Mehrere Pheromone und Instanzen	19
3.5 Agentenmodell	20
3.5.1 Bewegung	20
3.5.2 Pheromon-Interaktion	22
3.5.3 Ressourcen-Suche	24
3.5.4 Krankheitsübertragung	26
3.6 Initial-Zustand des Simulationsmodells	27
3.7 Technische Umsetzung	28
<b>4 Ergebnisse</b>	<b>31</b>
4.1 Versuchsaufbau	31
4.2 Betrachtung der Krankheitsausbreitung	32
4.3 Bewertungskriterien und Ausgangswerte	33
4.4 Strategien zur Eindämmung	34
4.5 Weitere Experimente	37
4.6 Anmerkungen	40
<b>5 Zusammenfassung</b>	<b>42</b>
5.1 Fazit	42
5.2 Ausblick	42
<b>Literaturverzeichnis</b>	<b>44</b>

---


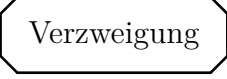






<b>Abbildungsverzeichnis</b>	<b>49</b>
<b>Tabellenverzeichnis</b>	<b>50</b>
<b>Algorithmen-Verzeichnis</b>	<b>51</b>
<b>Anlagenverzeichnis</b>	<b>52</b>
<b>Anlagen</b>	<b>53</b>
<b>Selbstständigkeitserklärung</b>	<b>69</b>

## Symbolverzeichnis

Im Folgenden sollen in der Arbeit genutzte Symbole erklärt werden. Die Listen sind nicht notwendigerweise vollständig. In [Tabelle 0.1](#) sind mathematische Symbole gegeben, die im Allgemeinen selten oder unterschiedlich benutzt werden. [Tabelle 0.2](#) gibt einen Überblick zu in Diagrammen genutzten Darstellungen. Diese wird so in der gesamten Arbeit eingehalten. Die Parameter der in [Abschnitt 3](#) beschriebenen Simulation wurden alle nach dem in [Tabelle 0.3](#) einseharen Schema benannt. Hierbei wurden die meisten Bezeichnungen von passenden englischen Wörtern abgeleitet, um eine kohärente Benennung mit der Simulations-Implementierung zu erreichen.

Symbol	Definition	Beispiel
$\leftarrow$	Weist einer Variable einen bestimmten Wert zu. Dieser Operator wird genutzt, um Verwechslungen zwischen der Bedeutung von $=$ in der Mathematik und in Programmiersprachen zu vermeiden.	$x \leftarrow 1$
$//$	Ganzzahlige Division: $// : \mathbb{Z} \times \mathbb{Z} \setminus \{0\} \rightarrow \mathbb{Z}$ $a//b \mapsto \frac{a - (a \bmod b)}{b}$	$5//2 = \frac{4}{2} = 2$
$\mathbb{N}$	Natürliche Zahlen <b>ohne</b> 0	$0 \notin \mathbb{N}$
$\mathbb{N}_0$	Natürliche Zahlen <b>mit</b> 0	$0 \in \mathbb{N}_0$
$\mathbb{E}$	Die geraden Zahlen	$\{-2, 4, 0\} \subset \mathbb{E}$
$\mathbb{O}$	Die ungeraden Zahlen	$\{-1, 3, 5\} \subset \mathbb{O}$

**Tabelle 0.1:** In der Arbeit genutzte mathematischen Symbolen, die im Allgemeinen nur selten oder unterschiedlich verwendet werden. Es sind jeweils kurze wörtliche Definitionen und Minimalbeispiele gegeben.

Darstellung	Beschreibung
 Zustand	Zustände in Flussdiagrammen
 Verzweigung	Konditionelle Verzweigungen in Flussdiagrammen
 Aktion	Auszuführende Aktionen, bzw. Ereignisse in Flussdiagrammen
 $r_1$	Darstellung von Agenten allgemein. <i>SEIR</i> -Zustand hier nicht relevant.
 $r_s$	Ein Agenten im <i>SEIR</i> -Zustand: <i>Susceptible</i>
 $r_e$	Ein Agenten im <i>SEIR</i> -Zustand: <i>Exposed</i>
 $r_i$	Ein Agenten im <i>SEIR</i> -Zustand: <i>Infectious</i>
 $r_r$	Ein Agenten im <i>SEIR</i> -Zustand: <i>Recovered</i>

**Tabelle 0.2:** Darstellung von den in dieser Arbeit genutzten Diagramm-Elementen inklusive kurzen Beschreibungen zu ihrer Verwendung. Bei mehrdeutigen Darstellung ist die Verwendung explizit in der Abbildungsbeschreibung gegeben oder aus dem Kontext ersichtlich.

Symbol	Beschreibung(en)	Abgeleitet von:
$c$	Anzahl	<i>count</i>
$h$	Höhe	<i>height</i>
$l$	Länge	<i>length</i>
$p$	Wahrscheinlichkeit, Position	<i>probability,</i> <i>position</i>
$r$	Radius	<i>radius</i>
$s$	Größe	<i>size</i>
$t$	Zeit	<i>time</i>
$v$	Geschwindigkeit	<i>velocity</i>
$w$	Breite	<i>width</i>
$\theta$	Winkel	-
$x$	Übrige Variablen	-

**Tabelle 0.3:** Übersicht zur Namensgebung der Simulationsparameter.

# 1 Einleitung

Seit jeher gehört die Eindämmung von Seuchen zu den großen Herausforderungen der Menschheit. Infektionskrankheiten haben weitreichende Auswirkungen auf die medizinische, soziale und wirtschaftliche Lage einer Gesellschaft. Definitionsgemäß handelt es sich bei Epidemien um zeitlich begrenzte Ausbrüche einer Erkrankung. Endemien sind geografisch begrenzt und Pandemien sind Epidemien großen Ausmaßes. Die jüngste Pandemie - der Ausbruch von *SARS-CoV-2* - hat verdeutlicht<sup>1</sup>, dass die schnelle Entwicklung von wirkungsvollen Strategien zur Eindämmung von Epidemien entscheidend ist, um individuelles Leid zu reduzieren, die Belastung von Gesundheitssystemen zu minimieren und wirtschaftliche Schäden zu begrenzen [2–4].

Epidemien sind schwer vorhersehbar und komplex zu modellieren. Traditionelle epidemiologische Modelle stützen sich auf mathematische und statistische Analysen, um die Ausbreitung von Infektionskrankheiten anhand von wenigen Parametern vorherzusagen. Durch die Ermittlung der Modellparameter können dann verschiedene Krankheitsausbreitungen analysiert und verglichen werden. Dieser Ansatz liefert wertvolle Erkenntnisse, baut aber auf zusammengesetzten Daten und statistischen Annahmen auf. Die Komplexität und Heterogenität von menschlichen sozialen Interaktionen können nicht erfasst und nur als Mittelwerte abgebildet werden.

An diesem Problem setzt die Bachelorarbeit an und nutzt anstelle einer abstrakten Modellierung des Infektionsgeschehens ein Modell einzelner Interaktionen. Es wird ein Multi-Agenten-System eingesetzt, das es ermöglicht, die Epidemie aus der Perspektive einzelner Individuen zu beschreiben. So können Entscheidungsprozesse berücksichtigt werden, die von Umweltfaktoren und spezifischen Interaktionen mit anderen Individuen beeinflusst werden.

## 1.1 Zielsetzung

Das Ziel dieser Bachelorarbeit liegt übergeordnet darin, ein Verständnis über die Wirksamkeit verschiedener Strategien zur Eindämmung von Epidemien zu gewinnen. Hierzu werden aufeinander aufbauend folgende Ziele gesetzt:

1. Die Entwicklung und Modellierung eines Multi-Agenten-Systems, das durch individuelle Entscheidungsprozesse der einzelnen Agenten eine realistische Ausbreitung einer Infektionskrankheit beschreibt.
2. Die Ergänzung des Modells durch eine von den Agenten gemeinsam zu lösende Aufgabe. Diese soll die Messung der Produktivität der Gesamtpopulation und hierdurch die Abbildung des wirtschaftlichen Schadens eines Infektionsgeschehens ermöglichen.
3. Das vollständige Modell soll als Simulation implementiert werden. Diese muss effizient laufen und Ausgabewerte als Mittel von mehreren Durchläufen berechnen, um statistisch belastbare Ergebnisse zu generieren.
4. Strategien zur Eindämmung des Infektionsgeschehens sollen entwickelt, in der Simulation getestet, analysiert und miteinander verglichen werden. Hierbei soll auf den Grad der Eindämmung, den die Strategien erreichen, und den entstandenen wirtschaftlichen Schaden eingegangen werden.

---

<sup>1</sup>Spätestens seit der Erklärung zur Pandemie durch die *WHO* am 11.03.2020 [1] wurden die Gefahr der Pandemie und notwendige Maßnahmen stark diskutiert.

## 1.2 Abgrenzung

Um die Realisierbarkeit der gesetzten Ziele zu gewährleisten, sind Abgrenzungen notwendig.

Hervorzuheben ist hierbei als erstes die Art der Modelle und der Simulationen. Das Verhalten der einzelnen Agenten wird nur abstrakt und verallgemeinert implementiert. Es werden grundlegende Aspekte wie die Bewegung und Krankheitsübertragung betrachtet. Komplexere Zusammenhänge, wie zum Beispiel die Rückkehr von Agenten in einzelne Wohnräume<sup>2</sup>, werden nicht betrachtet, da diese die Modellierung unnötig verkomplizieren und eine Interpretation der Ergebnisse erschweren.

Die Implementierung erfolgt als rein virtuelle *Software*-Simulation. Für eine genauere Betrachtung der Vorteile dieser Art der Simulation siehe [Abschnitt 2.7](#); eine Umsetzung auf physischen Robotern soll nicht betrachtet werden. Dennoch wird die Simulation an die Forschungsumgebung für die DeziBots an der Professur für Schwarmintelligenz und Komplexe Systeme (Universität Leipzig) angelehnt (siehe [Abschnitt 2.6](#)), um zu ermöglichen, dass an einem späteren Zeitpunkt Experimente in dieser Umgebung durchgeführt werden können.

Die in [Abschnitt 4](#) gewählten Eindämmungsstrategien sind einfach gehalten und orientieren sich an den Kernaspekten der simulierten Agenten, wie zum Beispiel den Bewegungsmustern. Strategien, die in der echten Welt zum Einsatz kommen, um Infektionskrankheiten einzudämmen, werden nicht betrachtet. Hier seien beispielhaft die Quarantäne von infizierten Individuen und Impfungen genannt. Diese Strategien sind komplex zu simulieren, da sie die Integration komplett neuer Aspekte in der Simulation voraussetzen. Für anknüpfende Arbeiten sind diese Strategien allerdings interessant. In dieser Arbeit wird eine begrenzte Anzahl an Eindämmungsstrategien betrachtet.

Bei der Erstellung von Multi-Agenten-Systemen und Simulationen liegen viele Parameter vor. Die Auswahl und Optimierung verschiedener Werte für alle Parameter ist an dieser Stelle nicht möglich. Für die meisten Experimente werden die Werte festgelegt und es wird versucht, diese Auswahl so zu treffen, dass sie die Ergebnisse nicht verzerren und sich an der echten Welt sinnvoll orientieren. Die Analyse der Ergebnisse ist im Kontext der gewählten Parameter zu betrachten und nicht verallgemeinerbar.

## 1.3 Aufbau der Arbeit

Zu Beginn der Arbeit werden in [Abschnitt 2](#) alle wichtigen Grundlagen erläutert, die für das Verständnis der Arbeit notwendig sind. Hierbei ist insbesondere die Erläuterung zum *SEIR*-Modell hervorzuheben, da sich das Infektionsmodell der Arbeit stark an diesem orientiert.

Anschließend werden in [Abschnitt 3](#) die verwendeten Methoden beschrieben. Es wird in erster Linie das genutzte Multi-Agenten-System schrittweise herausgearbeitet und zusammengesetzt. Anschließend wird die Umsetzung als Simulation erklärt.

---

<sup>2</sup>Dies wäre ein Aspekt, der in der echten Welt relevant sein kann. Speziell die Krankheitsübertragungen in einem Haushalt kann bei der Betrachtung von reellem Infektionsgeschehen nicht außer Acht gelassen werden. Allerdings öffnen solche Zusammenhänge die Diskussion für weitere Eigenschaften, wie hier zum Beispiel Familienzusammenhänge von Individuen und Besuche zwischen Haushalten. Dies soll hier verhindert werden.

In [Abschnitt 4](#) werden die erarbeiteten Methoden genutzt, um Simulationsexperimente durchzuführen. Hierzu wird als erstes der allgemeine Versuchsaufbau und die Wahl der Simulationsparameter beschrieben. Die Krankheitsausbreitung des Modells wird beispielhaft getestet und validiert. Anschließend werden Ausgangswerte ermittelt und auf deren Basis Bewertungskriterien definiert. Diese werden genutzt, um die folgenden Experimente zu bewerten. Die Diskussion der Ergebnisse erfolgt jeweils im direkten Kontext mit den Experimenten und nicht separat.

Zum Schluss werden in [Abschnitt 5](#) die Ergebnisse zusammengefasst wiedergegeben. Wie beschrieben, gibt es viele Möglichkeiten, das Modell zu erweitern und weitere Experimente durchzuführen. Hierzu werden in [Abschnitt 5.2](#) verschiedene Ausblicke auf mögliche zukünftige Untersuchungen aufgeführt.

In der Arbeit werden verschiedene mathematische Symbole und Darstellungen verwendet. Eine Übersicht der wichtigsten von diesen kann im [Symbolverzeichnis](#) am Anfang der Arbeit gefunden werden.

Im Anhang sind ausführliche Ausarbeitungen, die nicht unmittelbar zum Verständnis der Arbeit notwendig sind, gesondert zusammengestellt (siehe [Anlagenverzeichnis](#)). Digitale Anlagen, die nicht in dieser Arbeit darstellbar waren, sind auf den *Servern* der Universität Leipzig zu finden. Ein Überblick dieser ist in [Anlage 1](#) gegeben.



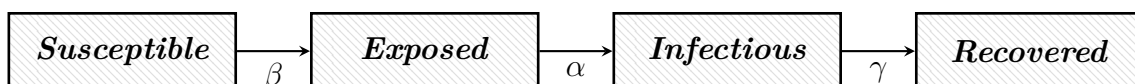
## 2 Grundlagen

In den folgenden Abschnitten wird der wissenschaftliche Hintergrund der Arbeit erläutert. Es werden Begriffe definiert, Erkenntnisse dargestellt und genutzte Modelle eingeführt.

### 2.1 *SEIR*-Modell

Als Basis für die Simulation von Krankheitsausbreitung wird ein epidemiologisches Modell genutzt, das die Gesamtpopulation in Gruppen einteilt. Das bekannteste solcher Modelle ist das 1927 von *Kermack* und *McKendrick* vorgestellte *SIR*-Modell [5]. In diesem wird die Population in drei Gruppen unterteilt<sup>3</sup>: *S* für *Susceptible*, *I* für *Infectious* und *R* für *Recovered* (bzw. *Removed*). Die Gruppe *S* stellt den Teil der Population dar, der mit einer gegebenen Krankheit noch nicht infiziert war und zukünftig infiziert werden kann. *I* ist die Gruppe der Individuen, die andere infizieren können und *R* beinhaltet alle, die eine Infektion durchlebt haben. Hierbei bedeutet ein Übergang eines Individuums in die Gruppe *R*, das Ausscheiden aus dem Infektionsgeschehen. Zwischen einer Heilung oder einem Todesfall nach der Infektion wird nicht unterschieden. *Kermack* und *McKendrick* zeigten die Effektivität des Modells zur Modellierung eines Pestausbruchs in Mumbai [5]. Es wird in der Epidemiologie [6, 7], für die Ausbreitung von *Computer*-Viren [8] und bei anderen Ausbreitungsgeschehen [9] angewandt.

Es existieren Anpassungen und Erweiterungen des *SIR*-Modells [10, 11]. Für diese Arbeit wird das *SEIR*-Modell genutzt, welches das *SIR*-Modell um eine Gruppe *E* für *Exposed* erweitert, die zwischen *S* und *I* liegt. Die Individuen gehen bei einer Infektion aus *S* in *E* und sobald sie andere Individuen anstecken können, in *I* über. Somit kann eine Inkubationszeit modelliert werden [12]. Alle Übergänge der einzelnen Individuen erfolgen linear in eine Richtung nach dem in [Abbildung 2.1](#) dargestellten Schema.



**Abbildung 2.1:** Darstellung zum Aufbau des *SEIR*-Modells. Die Übergänge von Individuen zwischen diesen sind als Pfeile zwischen den Gruppen eingezeichnet.

Für ein spezielles *SEIR*-Modell sei die Größe der Gesamtpopulation als  $N \in \mathbb{N}_0$  gegeben. Der Startzeitpunkt der Betrachtung ist  $t_0 = 0$ . Die Größen der einzelnen Gruppen sei im Verhältnis zur Zeit als Funktion gegeben (Beispielhaft  $S : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ ). Hierbei ist  $N$  konstant und durch die Gruppen komplett abgedeckt. Es gilt:

$$S(t) + E(t) + I(t) + R(t) = N \quad \forall t \in \mathbb{R}^+ \quad (1)$$

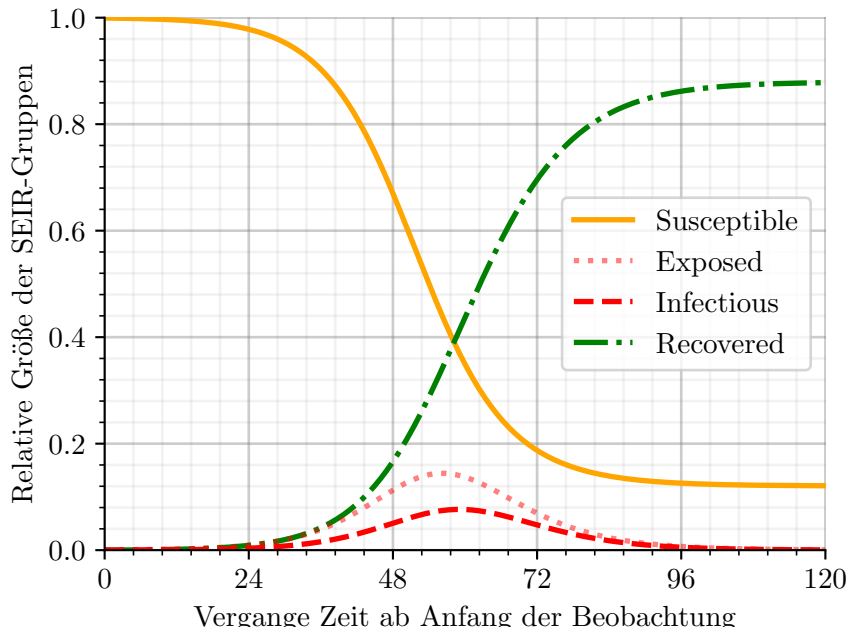
Zusätzlich sind die Transmissionsrate  $\beta$ , die Inkubationsrate  $\alpha$  und die Genesungs-/Sterberate  $\gamma$  gegeben. Diese beschreiben die Übergangsraten zwischen den einzelnen Gruppen (siehe [Abbildung 2.1](#)). Das Infektionsgeschehen wird durch folgendes System von Differentialgleichungen modelliert<sup>4</sup>:

$$\begin{aligned} S'(t) &= -\beta \cdot \frac{S(t) \cdot I(t)}{N} & E'(t) &= \beta \cdot \frac{S(t) \cdot I(t)}{N} - \alpha E(t) \\ I'(t) &= \alpha E(t) - \gamma I(t) & R'(t) &= \gamma I(t) \end{aligned} \quad (2)$$

<sup>3</sup>Der Name des Modells lässt sich von diesen Gruppen ableiten. In manchen Quellen wird es als *Kermack-McKendrick*-Modell bezeichnet.

<sup>4</sup> $S' : \mathbb{R}^+ \rightarrow \mathbb{R}$  stellt hierbei die erste Ableitung von  $S$ , die Veränderung von  $S$  zu einem gegebenen Zeitpunkt dar. Das gleiche gilt für  $E', I', R'$ .

Für gegebene Raten und Initialwerte  $S(0)$ ,  $E(0)$ ,  $I(0)$ ,  $R(0)$ , lässt sich dieses System lösen und es können Funktionszuweisungen für  $S$ ,  $E$ ,  $I$  und  $R$  bestimmt werden. Ein Beispiel einer solchen Lösung ist graphisch in [Abbildung 2.2](#) dargestellt.



**Abbildung 2.2:** Beispielhafte *SEIR*-Graphen, die durch Lösung der Differentialgleichungen berechnet wurden. Hierbei wurden folgende Parameter gewählt:  $N = 83.200.000$ ,  $\alpha = 2/11$ ,  $\beta = 4/5$ ,  $\gamma = 1/3$ ,  $E(0) = 40.000$ ,  $I(0) = 10.000$ ,  $R(0) = 0$ ,  $S(0) = 83.150.000$ . Die Größen der Gruppen sind auf der  $y$ -Achse verhältnismäßig zu  $N$  dargestellt. Der Betrachtungszeitraum ist  $[0, 120]$ . Diese Werte sind einem Beispiel vom [deutschen Wikipedia-Eintrag zum \*SEIR\*-Modell](#) (letzte Öffnung: 18. Mai 2023) nachempfunden.

In dieser Arbeit wird das *SEIR*-Modell genutzt, um die Zustände einzelner Individuen in der Krankheitsausbreitung darzustellen und auszuwerten (siehe [Abschnitt 3.5.4](#)).

## 2.2 Multi-Agenten-Systeme (*MAS*)

Agenten basierte Modellierung ist eine weitverbreitete und variantenreiche Methode zur Darstellung von Systemen mit mehreren Individuen [13]. Einzelne Individuen werden als Agenten bezeichnet, die sich in einer vorgegebenen Umgebung (z.B: ein Rechteckiges Spielfeld wie in [Abschnitt 3.2](#)) aufhalten. Die Eigenschaften eines Agenten sind unterschiedlich und können sich stark unterscheiden. Es gibt Vorschläge, dass alles, inklusive der Umgebung, in der sich Individuen aufhalten, als Agenten aufgefasst werden kann [14].

Hierbei sind die Modelle von natürlichen Schwärmen, wie zum Beispiel Bienen [15] oder Ameisen [16] inspiriert. Diese biologischen Schwärme bestehen aus Individuen, die simple Entscheidungen treffen. Insgesamt kann ein solcher Schwarm effizient komplizierte Verhaltensmuster ausprägen. Ein gutes Beispiel ist die Futtersuche von Ameisen [17].

Die Bezeichnung Multi-Agenten-System (*MAS*) wird in der Forschung zu Künstlicher Intelligenz genutzt, um Agenten basierte Modelle zu bezeichnen. Hierbei ist es das Ziel, wie bei den biologischen Schwärmen, komplexe Probleme zu lösen, die ein einzelnes Individuum überfordern [18]. Dies hat die Vorteile, dass das Verhalten der einzelnen Agenten einfacher

zu erklären und effizient umsetzbar ist. Somit können potenzielle Kosten reduziert und eine gute Skalierbarkeit durch das Hinzufügen von Agenten sichergestellt werden [18]. Aus diesen Gründen sind *MAS* nicht nur in der Forschung interessant, sondern finden auch Anwendung in der Industrie [19].

Bei dem in [Abschnitt 3](#) aufgebauten Modell handelt es sich um ein Multi-Agenten-System. Hierbei werden die aktiv mit ihrer Umwelt interagierenden Individuen als Agenten bezeichnet. In der Implementierung des Modells in [Abschnitt 3.7](#) wird diese Auslegung zur effizienten Umsetzung erweitert, in dem zum Beispiel Ressourcen-Quellen als Agenten aufgefasst und implementiert werden.

### 2.3 *Correlated Random Walk (CRW)*

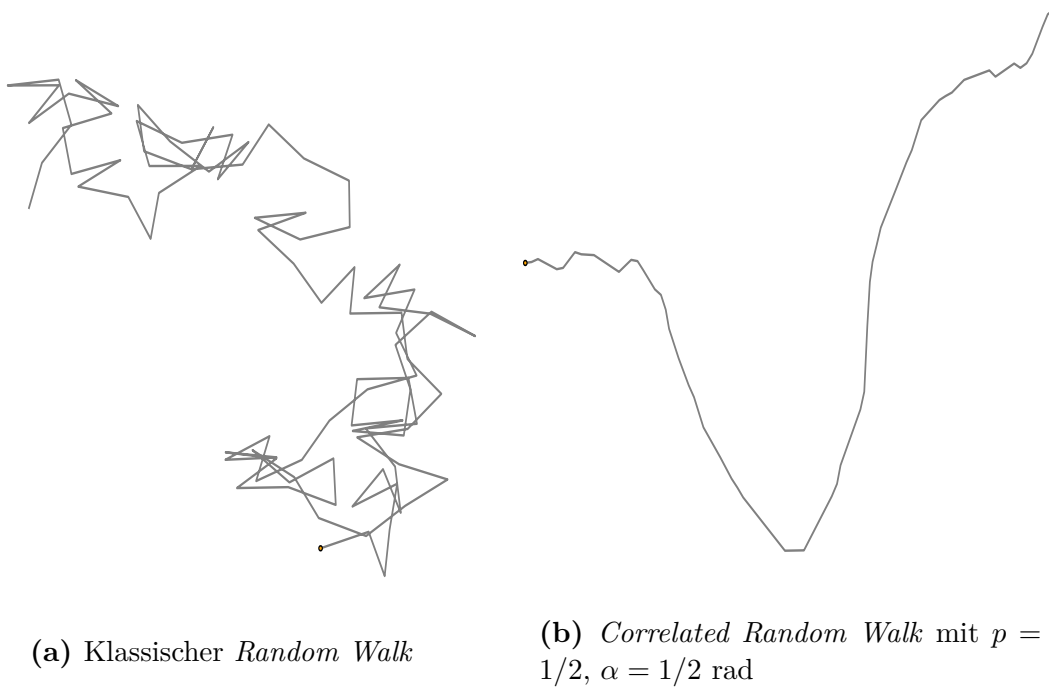
Als Basis für die Bewegung der Agenten zur Exploration wird ein *Random Walk* genutzt. Hierbei ist das einfachste Modell der klassische *Random Walk* auf einem eindimensionalen<sup>5</sup> Zahlenstrahl  $\mathbb{R}^1$  [20]. Hierbei startet ein Partikel<sup>6</sup> an einem Punkt auf dem Zahlenstrahl (z.B: 0) und bewegt sich in jedem Zeitschritt mit einer Wahrscheinlichkeit von  $\frac{1}{2}$  Richtung  $+\infty$ , bzw. mit der gleichen Wahrscheinlichkeit Richtung  $-\infty$ . Die Größe der Bewegung ist konstant. Das Modell lässt sich auf zwei Dimensionen erweitern, in dem sich das Partikel in jedem Schritt in eine zufällig gewählte Richtung bewegt. Das Resultat ist eine zufällige und zeitlich voneinander unabhängige Folge von Bewegungen (siehe [Abbildung 2.3a](#), für ein zweidimensionales Beispiel). Für die Bewegung von Tieren ist der klassische *Random Walk* kein realistisches Modell [21, 22], denn die Beobachtung, dass Tiere tendieren geradeaus in die gleiche Richtung zu laufen, wird nicht abgebildet [21].

Eine dahingehende Verbesserung stellt der sogenannte *Correlated Random Walk (CRW)* dar [21]. Dieser kann eindimensional beschrieben werden [23]: Ein Partikel startet mit einer zufällig gewählten Bewegungsrichtung (entweder  $+\infty$  oder  $-\infty$ ) und bewegt sich zu jedem Zeitschritt mit einer Wahrscheinlichkeit von  $p \in [0, 1]$  in die zuvor gewählte Richtung. Mit einer Wahrscheinlichkeit von  $1 - p$  ändert das Partikel zuerst die Richtung und bewegt sich anschließend. Es entsteht ein klarer zeitlicher Zusammenhang zwischen den Bewegungen. Wie viel sich ein Partikel geradeaus in eine Richtung bewegt, kann durch die Wahl von  $p$  beeinflusst werden. Dieses Modell kann auf zwei Dimensionen erweitert werden, in dem für jeden Zeitschritt sich das Partikel mit der Wahrscheinlichkeit  $p$  nach vorne bewegt oder um einen gegebenen Winkel  $\alpha$  nach links, bzw. rechts rotiert. In [Abbildung 2.3b](#) ist beispielhaft ein solcher Lauf dargestellt. Mit diesem Modell können die Nahrungssuche von Tieren [21] und einfachere Bewegungsmuster von Insekten [24] gut nachempfunden werden.

In [Abschnitt 3.5.1](#) wird eine angepasste Form des *CRW* genutzt, da er geeignet ist, die Bewegungsabläufe bei der Nahrungssuche zu simulieren. Dieser nutzt zufällige Rotationen in einem bestimmten Winkelbereich (z.B.  $[-\alpha, \alpha]$ ) statt Rotationen in festen Winkeln (wie hier  $\alpha$ ).

<sup>5</sup>Zur Reduktion der Komplexität und weil dies gut untersucht ist, wird hier von eindimensionalen Modellen gesprochen.

<sup>6</sup>Der Begriff Partikel wird hier genutzt, weil dieser ein einfaches Modell zur Darstellung von *Brownian Motion*, der Bewegung von Partikeln in einem Medium genutzt wird [20].



**Abbildung 2.3:** Beispielhafte Gegenüberstellung von klassischem *Random Walk* und *Correlated Random Walk* auf einem zweidimensionalen Feld.

## 2.4 Pheromone

Pheromone sind Botenstoffe, mit denen Tiere kommunizieren. Sie werden von einem Individuum abgegeben und lösen bei einem anderen eine spezifische Reaktion aus [25]. Die Simulation von Pheromonen spielt eine bedeutende Rolle in der wissenschaftlichen Forschung und im speziellen bei der Modellierung von *MAS*. Das Ziel solcher Simulationen ist es, die komplexen Mechanismen der Pheromon-basierten-Kommunikation und die sozialen Interaktionen und Verhaltensmuster zu verstehen. Das gewonnene Wissen wird genutzt, um an Pheromone angelehnte Systeme zur effizienten Kommunikation zwischen künstlichen Agenten in Simulationen zu nutzen (siehe [Abschnitt 2.4.3](#)).

### 2.4.1 Biologischer Hintergrund

Der biologische Hintergrund von Pheromonen bildet die Vorlage für die Simulation dieser in *MAS*. Daher ist es für diese Arbeit wichtig, die biologischen Grundlagen der Pheromon-freisetzung und -wahrnehmung zu verstehen. Pheromone sind in der Biologie chemische Botenstoffe, die zur Kommunikation innerhalb einer Spezies<sup>7</sup> genutzt werden [26]. Es gibt eine Reihe an Zwecken für diesen Informationsaustausch [25, 26]. Hervorzuheben sind die Suche nach Futter, die Kommunikation zur Fortpflanzung und zur Sicherheit (zum Beispiel durch die Warnung anderer Individuen vor Gefahren).

Die Botenstoffe werden typischerweise über Drüsen im Körper eines Organismus abgesondert. Bei Schuppenameisen liegen diese Drüsen primär am Hinterkörper der Individuen [27]. Die Abgabe erfolgt punktuell und lokal beschränkt.

<sup>7</sup>Es existieren auch Spezies, die Pheromone von fremden Spezies wahrnehmen. Dies stellt Ausnahmen dar und soll für diese Arbeit ignoriert werden.

Die Stoffe verbreiten sich über die Umgebung. Abhängig von der chemischer Zusammensetzung spielen verschiedene Mechanismen eine Rolle. Die meisten Pheromone sind flüchtige Substanzen. Die aus Verdunstung resultierenden gasförmigen Moleküle können sich über die Luft ausbreiten und über große Entfernungen verteilen [28]. Durch diesen Prozess wird die Anwesenheit der Stoffe zeitlich begrenzt. Die Verteilung erfolgt durch Diffusion [29] und die Konzentration wird durch Verwehungen und Verschleppungen anderer Individuen beeinflusst [28, 30, 31].

Die Stoffe können von anderen Individuen der selben Art wahrgenommen werden. Der Aufnahmemechanismus von den Substanzen unterscheidet sich abhängig von der Spezies und der chemischen Zusammensetzung des Pheromons. Im Allgemeinen kommen Chemozeptoren in den Sinnesorganen und spezialisierte Pheromonrezeptoren zum Einsatz [32]. Durch die Bindung des Pheromonmoleküls an einen entsprechenden Rezeptor wird ein Signal an das Gehirn, bzw. zentrale Nervensystem des Organismus gesendet und lösen eine Verhaltensreaktion aus [33].

Durch die kontrollierte Absonderung und die gegebenen Verhaltensreaktionen entstehen komplexere Verhaltensmuster der Gesamtpopulation. Beispielhaft seien hier die Futtersuche von Ameisen mittels Pheromonspuren [34] (siehe [Abschnitt 2.5](#)) und die Alarmierung bei Gefahren [35] genannt.

#### 2.4.2 Modellierung von Pheromon-Simulationen

Der Aufbau eines Anwendungsmodell erfordert die Definition eines Pheromonmodells. Hierzu ist es notwendig, die Eigenschaften der Pheromonproduktion, -freisetzung, -wahrnehmung und -ausbreitung zu beschreiben. All diese Aspekte hängen vom genutzten Botenstoff und den zu modellierenden Individuen ab. In vielen Anwendungen können Teile des Modells stark vereinfacht werden.

Dem Individuum steht eine bestimmte Menge an Pheromon zur Verfügung. Biologisch entspricht dies der Produktion von Drüsensekreten. Unter Berücksichtigung von Produktionsraten ist es möglich, realitäts-nahe Modell zu gestalten und Anwendungen von Pheromonen in der Natur zu erforschen. In vielen Anwendungsfällen, die nicht der biologischen Erforschung dienen, wird die Produktion nicht beachtet.

Bei der Modellierung der Pheromonfreisetzung werden die Mechanismen zur Steuerung der Abgabe des Botenstoffes eines Individuums festgelegt. Diese beeinflussen die resultierende Konzentration und die Position des Pheromons. Hierbei ist zu bestimmen, wie die Abgabe erfolgt (z.B. aktive Sekretion oder passives Ablegen ohne Krafteinwirkungen) und wo am Individuum die Stoffe freigesetzt werden (z.B. primär am Hinterkörper wie bei den Schuppenameisen in [Abschnitt 2.4.1](#)).

Die Pheromonwahrnehmung beinhaltet die Erfassung und grundlegende Verarbeitung der Botenstoffe. Somit kann dieser Prozess als Gegenstück zur Pheromonfreisetzung aufgefasst werden. Hier gilt es festzulegen, wie und wo die Pheromone erfasst werden. Es bietet z.B. die Art *Drosophila melanogaster* der Taufliegen-Familie einen Orientierungspunkt. Diese nehmen die Botenstoffe über sogenannte Antennen am Vorderkopf wahr [36].

Zwischen der Pheromonfreisetzung und -wahrnehmung muss die Ausbreitung der Pheromone über die Umgebung zeitlich modelliert werden. Es können verschiedene Aspekte Berücksichtigung finden. In vielen Fällen kommen Diffusionsmodelle zum Einsatz [30]. Diese können sicherstellen, dass freigesetzte Botenstoffe einen klaren lokalen Entstehungspunkt haben und sich über die Zeit in der Umgebung verteilen. Darüber hinaus hat die Modellierung des Abbaus / der Verdunstung des Pheromons einen großen Einfluss auf die Konzentration und beschränkt diese zeitlich.

### 2.4.3 Anwendungsmodelle für MAS

Die Anwendung von Pheromon-Simulationen in MAS bietet vielfältige Möglichkeiten zur Optimierung von kollektivem Verhalten und kooperativen Lösen von Herausforderungen. Die Pheromone dienen als Signale zwischen Agenten, um Kommunikation, Koordination und die Steuerung des Verhaltens zu ermöglichen. Hierbei gibt es zwei grundsätzliche Ansätze für die Durchführung der Pheromon-Simulationen:

1. Für **Software-Simulationen** wird das Pheromon-Modell in einer virtuellen Umgebung angewendet. In erster Linie können die finanziellen und zeitlichen Kosten für den Aufbau einer physischen Umgebung eingespart werden. Eine solche Simulation ist flexibler und ermöglicht eine bessere Skalierbarkeit. Es können Simulationen für verschiedene Parameter und große Agentenzahlen getestet werden.
2. Bei **Echtwelt-Simulationen** wird das Modell in einer physischen Umgebung aufgebaut. Dies bietet den Vorteil, dass realitätsnahe Experimente durchgeführt werden können. Komplexe Faktoren finden Berücksichtigung und können auf echte Anwendungsfelder übertragen werden. Während in Software-Simulation die Pheromone als abstraktes Konzentrationsfeld dargestellt werden können, ist hier die Bestimmung eines Übertragungsmediums erforderlich. Hierbei können zum Beispiel (wie in der Natur) chemische Botenstoffe wie Ethanol [37] oder phosphoreszierende Materialien (siehe Abschnitt 2.6) gewählt werden.

Vor dem Aufbau einer Echtwelt-Simulation kann allgemein ein zu untersuchende Konzept in einer Software-Simulation vorab validiert werden [37, 38].

Pheromon-Simulationen werden in der Forschung zu MAS angewandt. Hierbei spielen die Erforschung von Schwarmintelligenz und der Einsatz in der Robotik [37, 38] eine große Rolle. Der Nutzen für andere Felder, wie der Verkehrsplanung [39] und der Aufgabenverteilung [40], wird untersucht.

In dieser Arbeit wird ein abstraktes, stark vereinfachtes Pheromon-Modell für eine reine Software-Simulation genutzt. Eine detailliertere Beschreibung des Modells ist in Abschnitt 3.4 zu finden.

## 2.5 Pheromon basiertes Foraging

Eine der Hauptanwendungsbereiche von Pheromonen in der Natur und in MAS-Simulationen, ist die Nahrungssuche. Diese wird in der Forschung mit dem englischen Begriff *Foraging* bezeichnet.

Gut erforscht ist hierbei die Nahrungssuche von Ameisen. Diese nutzen Pheromon-Spuren um den Weg von ihrem Nest zu einer Nahrungsquelle und zurück zu markieren und andere Ameisen die Suche zu vereinfachen [41]. Stark vereinfacht kann dieses Verhalten durch folgenden Ablauf aus der Sicht einer einzelnen Ameise modelliert werden [41, 42]:

1. Die Ameise erkundet die Umgebung durch zufällige Bewegungen. Dies kann als ein *Random-Walk* (siehe Abschnitt 2.3) betrachtet werden.
2. Sobald eine Nahrungsquelle zufällig gefunden wird, nimmt die Ameise einen Teil der Nahrung auf.
3. Sie begibt sich auf den Rückweg zum Nest<sup>8</sup>. Hierbei legt die Ameise eine Pheromon-Spur hinter sich.

---

<sup>8</sup>Zur Findung des Rückwegs kommen in der Natur verschiedene Mechanismen zum Einsatz. Zum Beispiel können hier andere Pheromon-Spuren genutzt werden.



4. Am Nest angekommen, wird die Nahrung abgelegt und die Ameise macht sich erneut auf die Suche. Hierbei kann sie gefundenen Pheromon-Spuren folgen und schneller die Nahrungsquelle wiederfinden.

Durch das wiederholte Ablaufen eines Weges, zwischen dem Nest und einer Nahrungsquelle wird dieser verstärkt. Kürzere Wege setzen sich auf Dauer durch, da diese von Ameisen schneller und somit häufiger abgelaufen werden.

Dieses Verhalten wird für die Lösung verschiedenster Optimierungs-Probleme eingesetzt und als Ameisenalgorithmus bezeichnet [43]. Die Grundlage hierbei ist die Suche nach dem kürzesten Weg zwischen zwei Punkten durch den Algorithmus.

In dieser Arbeit wird das Konzept der Pheromon-Spuren, die den Standort von Ressourcen-Quellen markieren, für die Ressourcen-Suche der Agenten in [Abschnitt 3.5.3](#) genutzt. Hierbei kommen die Agenten ohne Nest aus und die Komplexität ihres Verhaltens ist stark vereinfacht.

## 2.6 DeziBot

Der Aufbau des Modells orientiert sich an den Eigenschaften des Roboters DeziBot<sub>3</sub> [44]. Die erste Generation des DeziBot wurde, mit Materialkosten von 7,71€, für den Einsatz an Schulen entwickelt [45]. Das Nachfolgemodell, der DeziBot<sub>2</sub> erweiterte die Plattform um signifikant leistungsfähigere *Hardware* und eine Vielzahl von Sensoren [46].

Die dritte Generation DeziBot<sub>3</sub> erlaubt die Massenproduktion und zusammengebaute Auslieferung der Roboter. Diese bestehen jeweils aus einer runden Hauptplatine, auf der ein *Arduino*-basierter Mikrocontroller, ein Akku, Sensoren und Aktuatoren angebracht sind. Der Roboter steht auf drei Stützen. Durch zwei Vibrationsmotoren können die beiden hinteren Stützen unterschiedlich stark bewegt werden. Eine Kombination ermöglicht die Fortbewegung und Richtungssteuerung des gesamten Roboters. In [Abbildung 2.4](#) ist ein DeziBot<sub>3</sub> zu sehen.

Der Zielanwendungsbereich des DeziBot<sub>3</sub> ist die schulische Bildung. Hier wird der Einsatz des Roboters erprobt und erforscht [44].

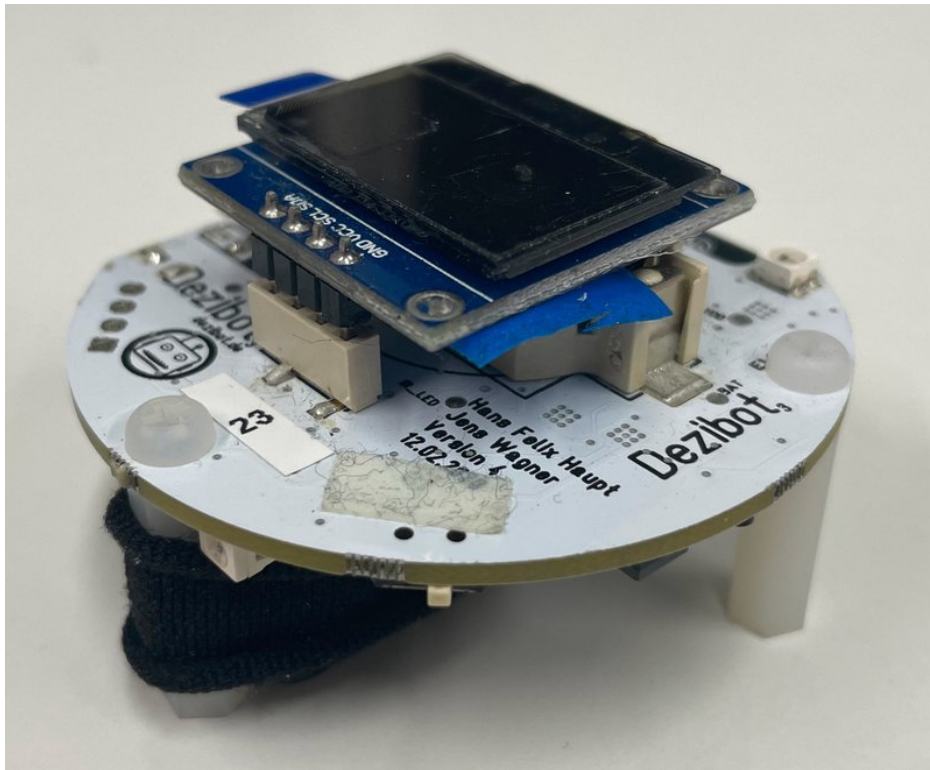
### 2.6.1 Anwendung in der Forschung

Die niedrigen Kosten des Roboters machen den DeziBot<sub>3</sub> attraktiv für die Forschung zur Schwarmintelligenz. An der Universität Leipzig forscht Tobias Jagla hierzu. Für diesen Zweck wird eine rechteckige Arena genutzt, die mit einer phosphoreszierenden Folie bespannt ist. Mithilfe von *UV-LEDs* können die Roboter die Folie bestrahlen. Dies ermöglicht die Echtwelt-Simulation von Pheromonfreisetzung. Über Lichtsensoren können die Roboter zuvor bestrahlte Stellen der Folie wahrnehmen. Der in [Abschnitt 3](#) beschriebene Modell-Aufbau orientiert sich stark an dieser Arena und dem DeziBot<sub>3</sub>.

## 2.7 Software-Simulation

Als *Software*-Simulation werden Simulationen bezeichnet, die in einer rein virtuellen Umgebung stattfinden und hierbei Vorteile gegenüber Echtwelt-Simulationen bieten. Großenteils überschneiden sich diese mit den in [Abschnitt 2.4.3](#) genannten Vorteilen für *Software*-Simulationen von Pheromon-Modellen. Hier sind die Vorteile im einzelnen aufgeführt:

1. **Beherrschung der Komplexität:** *MAS* können komplexe Interaktionsmuster aufweisen. In *Software*-Simulationen können diese durch Modelle auf die notwendigen Komponenten heruntergebrochen werden.



**Abbildung 2.4:** Ein DeziBot<sub>3</sub> in Seitenansicht. Der Roboter wurde für den Einsatz in der Forschung mit einem aufgesteckten *LED*-Bildschirm modifiziert.

2. **Flexibilität:** Die reduzierte Komplexität ermöglicht es, in verschiedenen Durchläufen der Simulation Parameter zu testen und Bedingungen zu verändern. Dies kann im Gegensatz zur Echtwelt-Simulation schnell und ohne großen Mehraufwand geschehen.
3. **Skalierbarkeit:** Die Simulation kann mit stark erhöhten Agentenzahlen und Umgebungsgrößen getestet werden. Dies ist teilweise eine Folge der beschriebenen Flexibilität.
4. **Kosteneffizienz:** Durch den Verzicht auf teure physische Ressourcen und eine umfangreiche Infrastruktur können *Software*-Simulationen in der Regel kosteneffizienter durchgeführt werden.
5. **Reproduzierbarkeit:** Durch die durchgängige, nicht kontrollierbare Änderung von Rahmenbedingungen sind Echtwelt-Simulationen schwer zu wiederholen. Bei *Software*-Simulationen ist dies anders. Die Rahmenbedingungen können kontrolliert und somit Versuchsdurchläufe reproduziert werden.

Zur Einhaltung des Umfangs dieser Arbeit ist ohnehin eine Reduktion der Komplexität der Modelle notwendig. Hier bietet eine *Software*-Simulation die notwendigen Kontrollmechanismen. Durch die in dieser Arbeit verwendete Reduktion kann auf die wesentlichen Forschungsziele fokussiert und diese von anderen Aspekten abgegrenzt werden.



## 3 Methoden

Im folgenden Abschnitt wird schrittweise der Aufbau der Simulation hergeleitet. Hierzu werden die einzelnen Bestandteile als mathematische Modelle eingeführt, die zusammengenommen als ein Simulations-Modell aufgefasst werden können. Folgend wird die technische Umsetzung des Modells beschrieben. Diese besteht aus der Implementierung und den Rahmenbedingungen für die Ausführung.

Alle aufgeführten Parameter für die Simulation können als vollständige Liste in den [Anlagen 2–10](#) eingesehen werden.

### 3.1 Allgemeines Modell

Die Simulation läuft in diskreten Zeitschritten ab. Hierbei wird im Zeitschritt 0 begonnen und die obere Grenze, der durchzuführenden Zeitschritte ist durch den Parameter  $t_{max} \in \mathbb{N}_0$  gegeben. Es ergibt sich somit die Menge der Zeitschritte als:

$$T = \{i \in \mathbb{N}_0 \mid i \leq t_{max}\} \quad (3)$$

Für jeden Zeitschritt  $i \in T$  führen alle Agenten in der Simulation ein *Update* aus (Der genaue Ablauf wird in [Abschnitt 3.5](#) beschrieben). Anschließend werden *Updates* des Pheromon-Modells (siehe [Abschnitt 3.4](#)) und des Ressourcen-Modells (siehe [Abschnitt 3.3](#)) ausgeführt.

Ereignisse des Modells sind teilweise wahrscheinlichkeitsbasiert. Zur Auswertung von Wahrscheinlichkeiten wird ein Zufallszahlen-Generator genutzt. Dieser produziert eine Folge von uniform verteilten Gleitkommazahlen im Intervall  $[0, 1)$ . Die genaue Folge ist abhängig vom *Seed*  $x_{random-seed} \in \mathbb{N}_0$ . Über die Variable  $x_{random} \in [0, 1)$  kann im Modell auf das jeweils nächste Element der Folge zugegriffen werden. Durch dieses System wird sichergestellt, dass die einzelnen Ereignisse stochastisch korrekt verteilt sind und die Simulation reproduzierbar und deterministisch ist. Im Speziellen liefert die Simulation bei gleichen Parametern und gleichem *Seed* die gleichen Ergebnisse.

### 3.2 Spielfeldmodell

Die Grundlage der Simulation ist die Umgebung  $U$ , in der sich die Agenten aufhalten. Diese wird hier als Spielfeld bezeichnet. Das Spielfeld ist ein Rechteck im kontinuierlichen Vektorraum  $\mathbb{R}^2$ , gegeben durch die Breite  $w_{arena} \in \mathbb{E} \cap \mathbb{N}$  und Höhe  $h_{arena} \in \mathbb{E} \cap \mathbb{N}$ . Das Zentrum des Spielfeldes ist der Koordinatenursprung  $(0, 0)$ . Der Aufbau ist in [Abbildung 3.1](#) dargestellt. Die Menge aller Positionen im Spielfeld ergibt sich als:

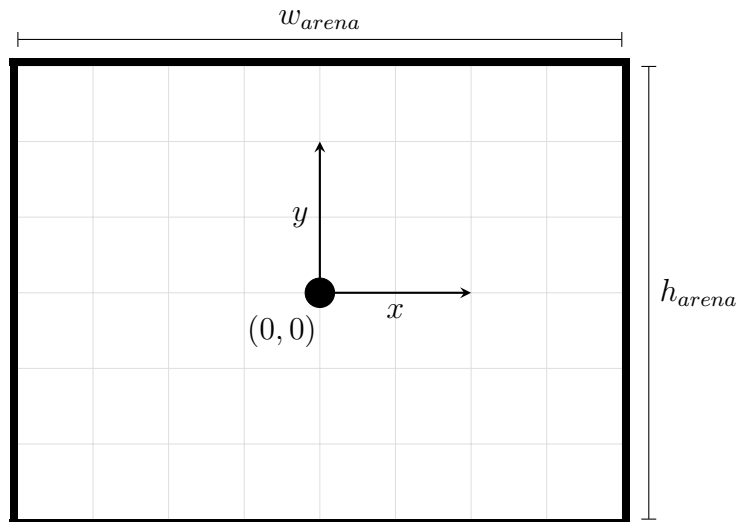
$$U = \left\{ (x, y) \in \mathbb{R}^2 \mid x \in \left[ -\frac{w_{arena}}{2}, \frac{w_{arena}}{2} \right) \wedge y \in \left[ -\frac{h_{arena}}{2}, \frac{h_{arena}}{2} \right) \right\} \quad (4)$$

Nennenswert ist hierbei, dass jeweils für die horizontalen und vertikalen Koordinaten, die untere Spielfeld-Grenze inklusive und die obere Spielfeld-Grenze exklusiv ist. Dies hängt mit technischen Einschränkungen der Simulationsumgebung zusammen. Für das Modell wird dies ignoriert. Den oberen Grenzen kann sich beliebig genähert werden.

Zur Ermittlung der Distanz zwischen zwei Punkten auf dem Spielfeld ist die Funktion  $\text{dist} : U^2 \rightarrow \mathbb{R}^+$  als euklidische Distanz gegeben<sup>9</sup>:

$$\text{dist}((x, y), (x', y')) = \sqrt{(x' - x)^2 + (y' - y)^2} \quad \forall (x, y), (x', y') \in U \quad (5)$$

<sup>9</sup>In den folgenden Abschnitten, wird die Distanz zwischen Elementen auf dem Spielfeld, wie zwei Agenten  $a_1, a_2 \in A$  teilweise verkürzt zu  $\text{dist}(a_1, a_2)$  anstelle von  $\text{dist}(\vec{p}_{a_1}, \vec{p}_{a_2})$ , wobei  $\vec{p}_{a_i}$  die Position von  $a_i$  repräsentiert.



**Abbildung 3.1:** Grundlegender Aufbau des Spielfeldes. Der Koordinatenursprung  $(0,0)$  ist mit Achsen eingezeichnet. Die Spielfeld-Ränder sind als schwarze Rechtecke dargestellt. Die Breite und Höhe des Spielfeldes sind am rechten und oberen Rand als Achsen gegeben. Die Wahl der Ausrichtung, ist irrelevant für das Modell. Die hier gezeigte Ausrichtung wird für die gesamte Arbeit beibehalten.

### 3.3 Ressourcenmodell

Zur Messung der Produktivität aller Agenten wird während der Simulation eine Aufgabe bewältigt. Diese besteht daraus, Ressourcen zu sammeln, welche in (Ressourcen-)Quellen gefunden werden können. Zu jedem Zeitpunkt der Simulation befinden sich  $c_{resources} \in \mathbb{N}_0$  Quellen in der Simulationsumgebung. Die Gesamtmenge dieser sei gegeben als  $R$ . Für den Zugriff auf die einzelnen Quellen sei für alle  $r \in R$  ihre Größe als  $s_r \in \mathbb{N}_0$  und Position als  $\vec{p}_r \in U$  gegeben.

Die Agenten können während eines Zeitschrittes einzelne Ressourcen aus den Quellen sammeln. Sobald dies geschieht, reduziert sich die Größe der Quelle um 1. Am Ende von jedem Zeitschritt, wird die *Update*-Funktion aus [Algorithmus 3.1](#) durchgeführt. Diese entfernt alle leeren Ressourcen-Quellen aus der Simulation und fügt jeweils eine neue Quelle mit der initialen Größe  $s_{resources} \in \mathbb{N}_0$  hinzu.  $R$  wird angepasst um diese Veränderungen zu reflektieren.

---

**Algorithmus 3.1** *Update*-Funktion für das Ressourcen-Modell. Die Funktion wird am Ende von jedem Zeitschritt ausgeführt, entfernt alle leeren Ressourcen und fügt jeweils neue hinzu.

---

```

1: function RESOURCEUPDATE()
2:   for  $r \in R$  do
3:     if  $s_r \leq 0$  then
4:       Entferne  $r$  aus der Simulation
5:        $R \leftarrow R \setminus \{r\}$ 
6:       Erstelle neue Quelle  $r'$  mit  $s_{r'} = s_{resources}$  und zufälliger Position  $\vec{p}_{r'} \in U$ 
7:        $R \leftarrow R \cup \{r'\}$ 
8:     end if
9:   end for
10: end function

```

---

### 3.4 Pheromonmodell

Auf Basis des Spielfeldes, wird ein Pheromon-Modell nach [Abschnitt 2.4.2](#) aufgebaut. Dieses fällt zur Reduktion der Komplexität und Verringerung der Laufzeit vereinfacht aus. Die größte Vereinfachung besteht aus den zeitlichen und räumlichen diskreten Aufteilungen des Modells. Dies ermöglicht die Einführung von mehreren Pheromonen, die unabhängig voneinander wirksam sind.

#### 3.4.1 Aufbau und Funktionsweise der Pheromone

Für die Erklärung der Funktionsweise wird hier von einem einzelnen Pheromon ausgegangen. Das Spielfeld  $U$  wird in Zellen der Größe  $1 \times 1$  eingeteilt und für jede Zelle eine Intensität und eine Zeit gespeichert. Diese Einteilung kann durch die Definition von zwei Matrizen vorgenommen werden:

$$\begin{aligned} \mathcal{I} &= (\mathcal{I}_{i,j})_{1 \leq i \leq h_{arena}, 1 \leq j \leq w_{arena}} \\ \mathcal{T} &= (\mathcal{T}_{i,j})_{1 \leq i \leq h_{arena}, 1 \leq j \leq w_{arena}} \end{aligned} \quad (6)$$

Zum Vereinfachten Zugriff auf die Werte der Matrizen mittels Positionen im Spielfeld werden folgende verkürzte Schreibweisen definiert:

$$\begin{aligned} i_{x,y} &= \mathcal{I}_{(y + \frac{h_{arena}}{2}), (x + \frac{w_{arena}}{2})} \\ t_{x,y} &= \mathcal{T}_{(y + \frac{h_{arena}}{2}), (x + \frac{w_{arena}}{2})} \end{aligned} \quad (7)$$

Hierbei stellt für eine gegebene Position  $(x, y) \in U$ , der Wert  $i_{x,y} \in \mathbb{R}$  die Intensität des Pheromons dar. Der Wert  $t_{x,y} \in T$  enthält, den Zeitschritt zu dem die gegebene Intensität gesetzt wurde.

Für das Pheromon sei eine Lebensdauer  $t_{lifetime} \in \mathbb{N}_0$  gegeben. Mittels dieser und dem aktuellen Zeitschritt der Simulation (ab hier gegeben als  $t_{current} \in T$ ) kann eine Verdunstung simuliert werden. Statt die Werte der Matrizen zu manipulieren, wird hier eine Zugriffsfunktion  $\varphi : U \cap \mathbb{Z}^2 \rightarrow \mathbb{R}$  für das Pheromon definiert, die für verdunstete Pheromone eine Intensität von 0 wiedergibt:

$$\varphi((x, y)) = \begin{cases} i_{x,y} & \text{falls } t_{current} - t_{x,y} \leq t_{lifetime} \\ 0 & \text{sonst} \end{cases} \quad (8)$$

Die Verdunstung erfolgt somit abrupt. Für  $t_{lifetime}$  Zeitschritte bleibt die Intensität an einer Stelle erhalten und fällt dann sofort auf 0.

Das Pheromon unterliegt keinerlei Diffusion. Alle Intensitätswerte bleiben aktuell, bis der Zeitpunkt zu dem der spezifische Wert gesetzt wurde, länger als  $t_{lifetime}$  zurückliegt oder der gegebene Wert überschrieben wird.

Für die Veränderung der Werte soll sichergestellt werden, dass alle Änderungen als erstes aufgenommen und zum Ende des Zeitschrittes ausgeführt werden. Hierzu sei eine einfache *Queue*  $X_{update-queue}$  mit den üblichen Standard-Definitionen für die Länge  $\text{len}(X_{update-queue})$ , für das Entnehmen eines Datums  $\text{pop}(X_{update-queue})$  und zum Einfügen von einem Datum  $\text{push}(X_{update-queue}, x)$  gegeben. Die Initialisierung der Veränderung eines Pheromon-Wertes kann mit der in [Algorithmus 3.2](#) gegebenen `SETPHEROMONE`-Funktion vorgenommen werden. Diese übergibt die Werte an  $X_{update-queue}$ . Für die restliche Arbeit sei folgende Kurzschreibweise über die Zugriffsfunktion definiert:

$$\varphi((x, y)) \leftarrow i \equiv \text{SETPHEROMONE}(x, y, i) \quad (9)$$

---

**Algorithmus 3.2** Funktionen zum setzen der Pheromon-Intensität  $i$  am Punkt  $(x, y)$  zum aktuellen Zeitpunkt  $t_{current}$  und zum *updaten* des Pheromons zum Ende des Zeitschrittes.

---

```

1: function SETPHEROMONE( $x, y, i$ )
2:   push( $X_{update-queue}, (x, y, i)$ )
3: end function

4: function PHEROMONEUPDATE
5:   while len( $X_{update-queue}$ ) > 0 do
6:     ( $x, y, i$ )  $\leftarrow$  pop( $X_{update-queue}$ )
7:      $i_{x,y} \leftarrow i$ 
8:      $t_{x,y} \leftarrow t_{current}$ 
9:   end while
10: end function

```

---

Es ist hervorzuheben, dass die SETPHEROMONE-Funktion sich auf das gegebene Pheromon bezieht. Dies ist für die Simulation mit mehreren Pheromonen wichtig.

Zum Ende von jedem Zeitschritt (nach den *Updates* der Agenten) wird die PHEROMONEUPDATE-Funktion für das Pheromon aufgerufen. Diese führt die Veränderungen in der Reihenfolge durch, in der sie zuvor initialisiert wurden. Diese kompliziertere Art der Veränderung von Pheromon-Werten hat zwei Vorteile:

- Es wird sichergestellt, dass alle Agenten für Ihr *Update* die gleichen Pheromon-Werte vorliegen haben. Somit hat kein Agent Vor-/Nachteile, weil dieser nach den anderen aktualisiert wurde.
- Bei der technischen Umsetzung wird das Modell parallelisiert und es muss sichergestellt werden, dass die verschiedenen Prozesse synchron laufen.  $X_{update-queue}$  ermöglicht einen Zugriff auf die einzelnen Veränderungen der Pheromon-Werte mit einer klaren Reihenfolge. Die Prozesse werden genauer in [Abschnitt 3.7](#) beschrieben.

Die gegebene Zugriffs-Funktion ist die Schnittstelle für alle Pheromon-Interaktionen der Agenten. Die genauen Abläufe der Pheromonproduktion, -freisetzung und -wahrnehmung sind im Agenten-Modell weiterführend beschrieben (siehe [Abschnitt 3.5.2](#)).

### 3.4.2 Mehrere Pheromone und Instanzen

Zur Definition von mehreren Pheromonen (hier beispielhaft zwei), seien getrennt voneinander einzelne Intensitäts- und Zeitschritts-Matrizen definiert. Aufbauend können die Zugriffs-Funktionen  $\varphi_1, \varphi_2$  über gegebene Lebenszeiten  $t_1, t_2 \in \mathbb{N}_0$  aufgebaut werden. Die Pheromone funktionieren unabhängig voneinander und die Zugriffs-Funktionen können ohne Beachtung der anderen Pheromone genutzt werden.

Zur Steuerung des Verhaltens der Agenten werden zwei Pheromon-Instanzen genutzt. Für diese seien entsprechende Matrizen gegeben.

Das erste Pheromon wird als Ressourcen-Indikator-Pheromon bezeichnet. Dieses wird von den Agenten genutzt, um gefundene Ressourcen-Quellen zu markieren und anderen Agenten bei ihrer Ressourcen-Suche zu unterstützen. Dieses Verhalten wird im späteren [Abschnitt 3.5.3](#) beschrieben. Hierzu sei die Zugriffs-Funktion  $resInd : U \cap \mathbb{Z}^2 \rightarrow \mathbb{R}$  mit der Lebenszeit  $t_{resource-indicator-lifetime} \in \mathbb{N}_0$  für das Pheromon gegeben.

Als zweites wird das sogenannte Warn- bzw. Warnungs-Pheromon eingesetzt. Dieses kann von Agenten, die infektiös und sich dessen bewusst sind, freigesetzt werden. In [Abschnitt 4.5](#) wird das Pheromon genutzt, um infektiöse Agenten zusammen zu führen, indem diese Spuren des Pheromons folgen. Die Zugriffs-Funktion  $warn : U \cap \mathbb{Z}^2 \rightarrow \mathbb{R}$  ist über die Lebenszeit  $t_{warning-lifetime} \in \mathbb{N}_0$  definiert.

### 3.5 Agentenmodell

Das Verhalten der einzelnen Agenten ist im folgenden Agenten-Modell erklärt. Hierzu sei die Gesamtmenge der Agenten  $A$  durch die Gesamtzahl der Agenten  $c_{agents}$  gegeben. Diese kann aus den gegebenen Parametern für die initialen Anzahlen von Agenten in den verschiedenen *SEIR*-Zuständen (siehe [Abschnitt 3.6](#)) ermittelt werden:

$$\begin{aligned} c_{agents} = & \quad c_{initial-susceptible} + c_{initial-exposed} \\ & + c_{initial-infectious} + c_{initial-recovered} \end{aligned} \quad (10)$$

Die Anzahl der Agenten bleibt über die gesamte Simulation konstant.

Die Agenten führen jeden Zeitschritt ein *Update* durch und haben in diesem die Möglichkeit auf ihre Umgebung, andere Roboter, Pheromone und Ressourcen zu reagieren. Entscheidend ist der eigene *SEIR*-Zustand (genaueres in [Abschnitt 3.5.4](#)) und der Ressourcen-Suche-Zustand (genaueres in [Abschnitt 3.5.3](#)).

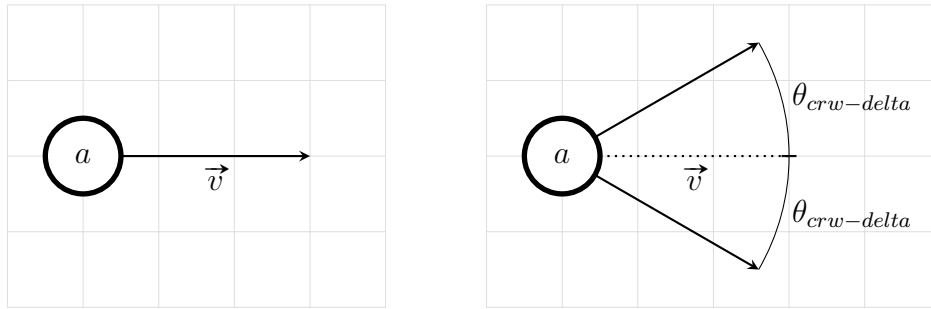
Zur Erörterung wird ab diesem Punkt großenteils ein Agent  $a \in A$  betrachtet. Für diesen sei seine Position  $\vec{p} = (p_x, p_y) \in U$  und seine vektorielle Geschwindigkeit  $\vec{v} = (v_x, v_y) \in \mathbb{R}^2$  gegeben.  $\vec{v}$  ist durch den Simulations-Parameter  $v_{agent}$  beschränkt. Es gilt:  $|\vec{v}| = v_{agent}$ .

#### 3.5.1 Bewegung

Als Basis-Bewegungs-Muster dient eine angepasste Version des in [Abschnitt 2.3](#) beschriebenen *CRW*. Hierzu entspricht  $\vec{v}$  zu Beginn eines jedem *Updates* der im letzten Zeitschritt gesetzten Bewegungsrichtung. Als  $p_{crw} \in [0, 1]$  ist die Wahrscheinlichkeit gegeben, dass  $r$  seine Geschwindigkeit beibehält und geradeaus läuft. Die Wahrscheinlichkeit wird mittels  $x_{random} < p_{crw}$  ausgewertet. Falls diese Auswertung positiv verläuft, wird nichts verändert. Falls die Auswertung negativ ist, wird  $\vec{v}$  um einen zufälligen Winkel von  $\theta \in [-\theta_{crw-delta}, \theta_{crw-delta}]$  rotiert. (Die Auswahl von  $\theta$  erfolgt mittels  $x_{random}$ . Somit ist  $\theta$  in  $[-\theta_{crw-delta}, \theta_{crw-delta}]$  uniform verteilt.) Am Ende wird der Agent ausgehend von  $\vec{p}$  entlang  $\vec{v}$  bewegt. Dieses Verhalten ist in [Abbildung 3.2](#) dargestellt.

Die Bewegung von  $a$  entlang  $\vec{v}$  bedeutet im Regelfall, dass die Position  $\vec{p}$  auf  $\vec{p} + \vec{v}$  gesetzt wird. Falls durch diese Bewegung der Agent das Spielfeld verlassen würde und gilt:  $(\vec{p} + \vec{v}) \notin U$ , muss die Bewegung angepasst werden. Hierzu wurden drei verschiedene Verfahren in Erwägung gezogen:

1. **Umbrechen:** Sobald  $a$  das Spielfeld verlässt, taucht er auf der gegenüberliegenden Seite des Spielfeldes auf und behält die aktuelle Bewegungsrichtung bei. Alle Agenten können sich unbegrenzt in eine Richtung bewegen. Das Spielfeld verhält sich, wie die Oberfläche von einem Torus.  
Dies entspricht nicht dem Spielfeld, in dem sich die *DeziBots* bewegen. Dieses Verhalten wird daher ausgeschlossen.



(a) Darstellung zu dem Fall, dass die zufällige Auswertung im *CRW* positiv ist und  $a$  geradeaus entlang  $\vec{v}$  bewegt wird.

(b) Darstellung zu dem Fall, dass die zufällige Auswertung im *CRW* negativ verlief und  $a$  seine Bewegungsrichtung  $\vec{v}$  verändert.

**Abbildung 3.2:** Basis-Bewegungs-Muster für die Bewegung der Agenten. Gezeigt ist ein Agent  $a$ , der sich nach dem *CRW* bewegt. Die Bewegungsrichtung von  $a$  ist als  $\vec{v}$  eingezeichnet und das Rotations-Delta für den *CRW* ist durch  $\theta_{crw-delta}$  gegeben.

2. **Strikte Spielfeld-Ränder:** Die Position wird auf das Spielfeld komponentenweise begrenzt, ohne Anpassungen vorzunehmen. Beispielsweise hier für die  $x$ -Koordinate gegeben durch:  $p_x \leftarrow \text{clamp}\left(p_x + v_x, -\frac{w_{arena}}{2}, \frac{w_{arena}}{2}\right)$ <sup>10</sup>. Die Bewegungsrichtung bleibt unverändert.

Das Resultat ist, dass Agent im Spielfeld gehalten werden. Sie reagieren nicht auf die Spielfeld-Ränder und bewegen sich gegen diese<sup>11</sup>.

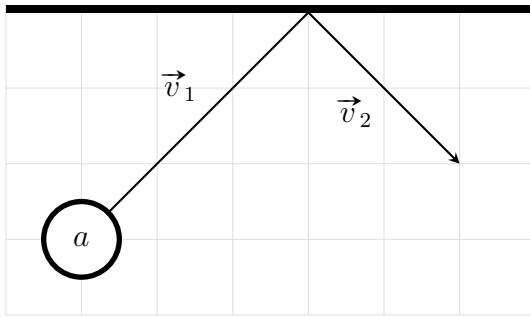
3. **Abprallen:** Hier wird die Position auf das Spielfeld begrenzt und die aktuelle Bewegungsrichtung angepasst. Dieses Verhalten stellt somit eine Anpassung der strikten Spielfeld-Ränder dar. Beim Treffen auf einen Spielfeld-Rand wird der Agent bis an den Rand bewegt und seine Bewegungsrichtung senkrecht zu der Achse des Spielfeld-Randes gespiegelt. Dies wird als „Abprallen“ bezeichnet. Anschließend wird der Agent entlang der neuen Bewegungsrichtung für die verbleibende Bewegungslänge bewegt (dargestellt in [Abbildung 3.3a](#)). Dieser Prozess wird iterativ wiederholt, bis der Agent  $v_{agent}$  Längeneinheiten weit bewegt wurde. Es kann zu mehreren Bewegungsänderungen in einem Agenten-*Update* kommen (siehe [Abbildung 3.3b](#)).

Insgesamt prallen die Agenten von den Spielfeld-Rändern ab und das gesamte Verhalten lässt sich gut mit dem Reflexionsgesetz: „Einfallswinkel gleich Ausfallswinkel“ zusammenfassen. Dieses Verhalten kann als eine vereinfachte Modellierung von Agenten, die Spielfeld-Ränder erkennen und diesen ausweichen, betrachtet werden.

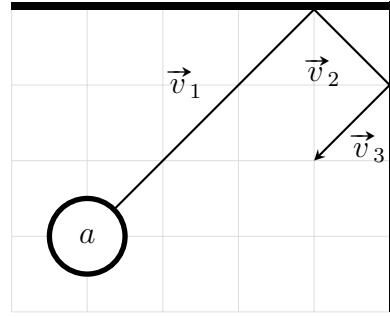
Für die Simulation wird das zuletzt beschriebene Verfahren genutzt. Mit Diesem ergibt sich das grundlegende Bewegungs-Verhalten von  $a$ .

<sup>10</sup>Wobei  $\text{clamp} : \mathbb{R}^3 \rightarrow \mathbb{R}$  gegeben sei durch  $(x, y, z) \mapsto \max(y, \min(z, x))$

<sup>11</sup>In Experimenten konnte weiterhin beobachtet werden, dass Agenten sich hierdurch an den Rändern sammeln.



(a) Darstellung für ein einfaches Abprallen von  $a$  am oberen Spielfeld-Rand. Die einzelnen Abschnitte der Bewegung von  $a$  sind als  $\vec{v}_1, \vec{v}_2$  gegeben, wobei gilt:  $v_{agent} = |\vec{v}_1| + |\vec{v}_2|$



(b) Darstellung für mehrfaches Abprallen von  $a$ , am oberen und am rechten Spielfeld-Rand. Die einzelnen Abschnitte der Bewegung von  $a$  sind als  $\vec{v}_1, \vec{v}_2, \vec{v}_3$  gegeben, wobei gilt:  $v_{agent} = |\vec{v}_1| + |\vec{v}_2| + |\vec{v}_3|$

**Abbildung 3.3:** Darstellung zum Verhalten der Agenten an den Spielfeld-Rändern. Die Größe der Längeneinheiten ist graphisch als Gitter dargestellt. Der Agent  $a$  ist eingezeichnet und beginnt das *Update* mit der Bewegungsgeschwindigkeit  $\vec{v} = (5, 5)$ .

### 3.5.2 Pheromon-Interaktion

Die Agenten können über die in [Abschnitt 3.4](#) gegebenen Pheromon-Modellen miteinander kommunizieren. Dies geschieht über die Zugriffs-Funktionen *resInd* und *warn*. In der Natur nutzen Lebewesen für verschiedene Pheromone einheitliche Interaktions-Mechanismen (z.B. werden verschiedene Botenstoffe von Schuppenameisen über deren Antennen am Vorderkopf wahrgenommen [47]). Hier werden die Interaktionen somit unabhängig von den Zugriffs-Funktionen, bzw. für eine abstrakte Zugriffs-Funktion  $\varphi$  definiert.

Einem Agenten stehen zur Reduktion der Komplexität zu jedem Zeitpunkt beliebig große Mengen an Pheromonen zur Verfügung. Die Produktion der Pheromone wurde bewusst außer acht gelassen.

Die Pheromonfreisetzung kann durch die Unbegrenztheit der Pheromone stark vereinfacht werden. Hierzu wird auf dem Pheromon-Gitter „gemalt“. Das bedeutet, dass ein Agent die Konzentrationen des Pheromons in einem bestimmten Feld überschreibt, ohne die Verteilung des Pheromons auf das Feld genauer zu spezifizieren. Für die Definition dieses sogenannten Schreibfeldes, seien die normalisierte Laufrichtung  $(\vec{v})$  von  $a$  und ein normalisierter Vektor, der vom Agenten nach links zeigt<sup>12</sup>  $(\vec{v}_{left})$  gegeben. Das Schreibfeld ist  $w_{pheromone-draw} \in \mathbb{N}_0 \cap \mathbb{O}$  Zellen breit und  $h_{pheromone-draw} \in \mathbb{N}_0$  Zellen lang. Es ergibt sich aus dem in [Algorithmus 3.3](#) gegebenen Algorithmus `DRAWFIELD( $n$ )`. Dieser gibt für eine Eingabe  $n \in \mathbb{N}_0$  mit  $n < w_{pheromone-draw} \cdot h_{pheromone-draw}$  jeweils die Punkte des Feldes in einer vordefinierten Reihenfolge wieder. Durch das Inkrementieren von  $n$  kann leicht über die Punkte iteriert werden. Die Funktion `DRAWPHEROMONE( $\varphi$ )` tut dies und nutzt die, sich ergebenden Positionen, um für eine gegebene Pheromon-Zugriffs-Funktion  $\varphi$  die Pheromon-Intensitäten an den Punkten in der Arena auf 1 zu setzen. Hierbei muss auf Grund der diskreten Auflösung von den Pheromonen die Position (ab-)gerundet werden.

Das sich ergebende Feld und die Iterations-Reihenfolge sind in [Abbildung 3.4a](#) dargestellt, wobei hier das Zentrum von jeder Zelle einen Punkt darstellt.

<sup>12</sup>Dieser kann aus  $\text{norm}(\vec{v})$  durch eine Rotation von  $90^\circ$  gegen den Uhrzeigersinn erreicht werden.



---

**Algorithmus 3.3** Algorithmen für das Schreiben von Pheromon-Intensitäten, für eine bestimmte Pheromon-Zugriffs-Funktion  $\varphi$  aus der Sicht, des Agenten  $a$ .

---

```

1: function DRAWFIELD( $n$ )
2:    $y \leftarrow n \bmod w_{\text{pheromone-draw}}$ 
3:    $x \leftarrow 1 + (n // w_{\text{pheromone-draw}})$ 
4:    $\vec{p}_{\text{mid}} \leftarrow \vec{p} - x \cdot \text{norm}(\vec{v})$ 
5:   if  $y = 0$  then
6:     return  $\vec{p}_{\text{mid}}$ 
7:   else if  $y \bmod 2 = 1$  then
8:     return  $\vec{p}_{\text{mid}} + \frac{y+1}{2} \cdot \text{norm}(\vec{v}_{\text{left}})$ 
9:   else
10:    return  $\vec{p}_{\text{mid}} - \frac{y}{2} \cdot \text{norm}(\vec{v}_{\text{left}})$ 
11:  end if
12: end function

13: function DRAWPHEROMONE( $\varphi$ )
14:  for  $n < w_{\text{pheromone-draw}} \cdot h_{\text{pheromone-draw}}$  do
15:     $(x, y) \leftarrow \text{DRAWFIELD}(n)$ 
16:     $\varphi(\lfloor x \rfloor, \lfloor y \rfloor) \leftarrow 1$ 
17:  end for
18: end function

```

---

Für die Pheromon-Wahrnehmung sei eine Funktion  $\text{READFIELD}(n)$  gegeben, die parallel zu  $\text{DRAWFIELD}(n)$  funktioniert. Hierzu müssen im Algorithmus die Variablen für die Breite und Höhe des Feldes durch  $w_{\text{pheromone-detect}} \in \mathbb{N}_0 \cap \mathbb{O}$  und  $h_{\text{pheromone-detect}} \in \mathbb{N}_0$  ersetzt werden. In Zeile 4 muss das Minus durch ein Plus ausgetauscht werden. Das sich ergebende Feld mit Breite  $w_{\text{pheromone-detect}}$  und Länge  $h_{\text{pheromone-detect}}$  liegt vor dem Agenten (siehe [Abbildung 3.4b](#)) und wird ab diesem Punkt als Lesefeld bezeichnet. Mit der Funktion  $\text{FIRSTPHEROMONE}(\varphi)$  aus [Algorithmus 3.4](#) kann die erste Position im Sichtfeld gefunden werden, an der ein Pheromon mit Intensität größer 0 liegt. Die Iterations-Reihenfolge stellt hierbei sicher, dass Pheromone die direkt vor dem Agenten liegen vor solchen gefunden werden, die seitlich vor ihm liegen.

---

**Algorithmus 3.4** Algorithmus für die Ermittlung der Position, des ersten Pheromons im Sichtfeld des Agenten  $a$ . Die Reihenfolge, der Pheromone ist durch  $\text{READFIELD}()$  gegeben und die Funktion nutzt eine beliebige abstrakte Pheromon-Zugriffs-Funktion  $\varphi$

---

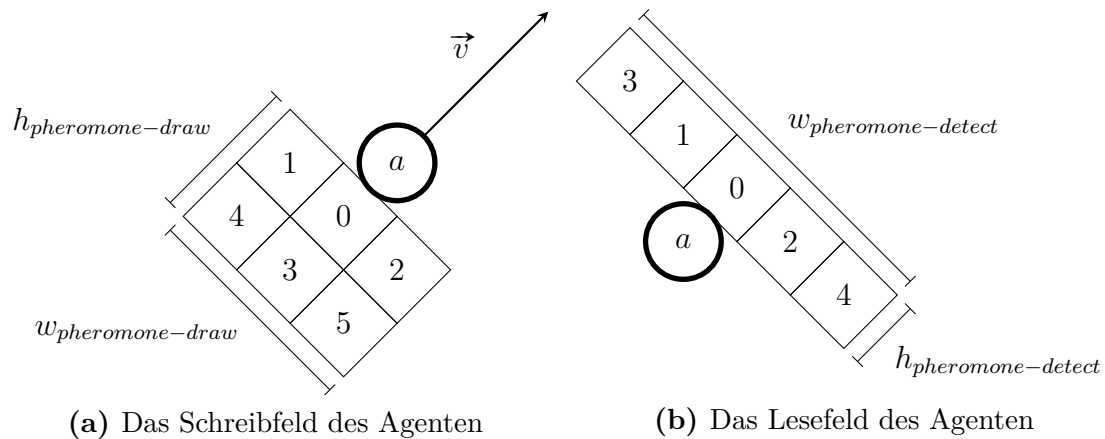
```

1: function FIRSTPHEROMONE( $\varphi$ )
2:  for  $n < w_{\text{pheromone-detect}} \cdot h_{\text{pheromone-detect}}$  do
3:     $(x, y) \leftarrow \text{READFIELD}(n)$ 
4:    if  $\varphi(\lfloor x \rfloor, \lfloor y \rfloor) \geq 0$  then
5:      return  $(x, y)$ 
6:    end if
7:  end for
8:  return NULL
9: end function

```

---





**Abbildung 3.4:** Darstellungen der Lese- und Schreibfelder eines Agenten für die Pheromon-Interaktion. Es ist jeweils ein Agent  $a$  mit der Position  $\vec{p} = (0, 0)$  und Bewegungsrichtung  $\vec{v} = (2, 2)$  gegeben. Die geplante Bewegung ist in [Abbildung 3.4a](#) als Pfeil eingezeichnet. Zusätzlich sind die Felder mit den Iterationsreihenfolge durchnummeriert.

Dieser Einteilung der Interaktion, wobei Agenten Pheromone vor sich wahrnehmen und hinter sich freisetzen ist von Schuppenameisen inspiriert. Diese nehmen Pheromone über Antennen am Vorderkopf wahr [47] und nutzen Drüsen am Hinterkörper für die Freisetzung [27]. Diese Grundlage eignet sich, da in der Simulation Pheromone als Spuren genutzt werden.

### 3.5.3 Ressourcen-Suche

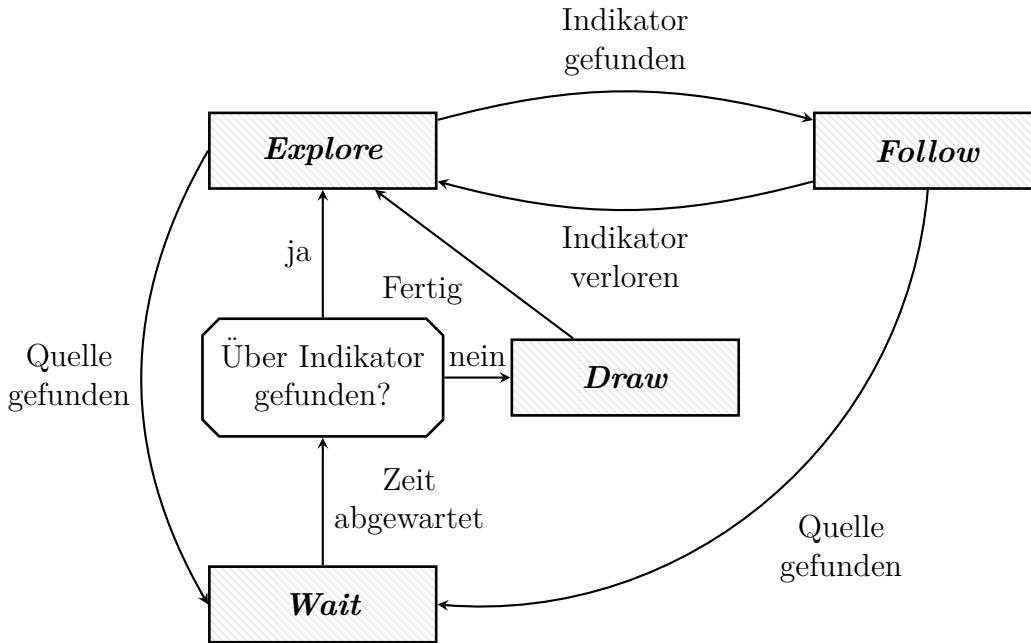
In [Abschnitt 3.3](#) wurde das grundlegende Modell zu den sich im Spielfeld befindenden Ressourcen-Quellen beschrieben. Die Aufgabe der Agenten soll es sein, diese Quellen zu finden und Ressourcen einzusammeln. Zur Erhöhung der Effizienz kommunizieren die Agenten über Pheromon-Spuren die Positionen der Quellen. Hierbei nutzen die Agenten ein Verhaltensmodell, dass an die Futtersuche von Ameisen angelehnt ist (siehe [Abschnitt 2.5](#)). Dies geschieht über die Pheromon-Zugriffs-Funktion  $resInd$ . Zur Vereinfachung des Modells müssen die Agenten die Ressourcen nur einsammeln und nicht zu einem Nest zurückbringen.

Das Verhaltensmodell beschreibt strukturell die Agenten als Automat. Hierzu befindet sich jeder Agent in einem von vier Zuständen, die durch die folgende Menge gegeben sind:

$$S = \{Explore, Follow, Wait, Draw\} \quad (11)$$

Die Übergänge zwischen den Zuständen sind als Flussdiagramm in [Abbildung 3.5](#) dargestellt. Die Verhaltensweisen der Agenten in den Zuständen werden in den Absätzen, an Hand von  $a$  einzeln erklärt. Hierbei wird jeweils davon ausgegangen, dass sich  $a$  in dem gegebenen Zustand befindet. Zusätzlich sind die Abläufe als Algorithmen in [Anlagen 12–15](#) aufgeführt.

Der *Explore*-Zustand repräsentiert die Erkundung der Agenten. In jedem Zeitschritt bewegt sich  $a$ , nach dem in [Abschnitt 3.5.1](#) gegebenen Basis-Bewegungsmuster. Anschließend werden die Ressourcen-Quellen durchsucht. Falls im Radius von  $r_{resource-detect} \in \mathbb{R}^+$  um  $a$  eine Quelle liegt, von der  $a$  noch nichts gesammelt hat, beginnt  $a$  Ressourcen einzusammeln. Hierzu wird in den Zustand *Wait* gewechselt. Falls keine Quelle gefunden wurde, sucht  $a$  nach einer Pheromon-Spur in seinem Sichtfeld. Falls diese Suche positiv verläuft, wird in den Zustand *Follow* übergegangen.



**Abbildung 3.5:** Allgemeine Übersicht zum Aufbau der Ressourcen-Suche von Agenten

Im *Wait*-Zustand sammelt  $a$  die Ressourcen von einer gefundenen Quelle ein und wartet nachfolgend  $t_{resource-collect} \in \mathbb{N}_0$  Zeitschritte. In dieser Zeit kann sich der Agent nicht bewegen. Der Zustand kann als Modellierung des Sammelprozesses und (stark vereinfacht) als Repräsentation des Abtransportes der Ressource betrachtet werden. Falls  $a$  die Quelle durch Erkundung gefunden hat<sup>13</sup> geht dieser nach der abgewarteten Zeit in *Draw* über, ansonsten in *Explore*.

Der *Draw*-Zustand stellt die Markierung von gefundenen Quellen dar.  $a$  malt von der Quelle ausgehend eine Linie in die Richtung, aus der er die Quelle gefunden hat. Hierzu dreht er sich beim Übergang in den Zustand um. Mit jedem Zeitschritt in diesem Zustand, geht  $a$  einen Schritt vorwärts und legt eine Pheromon-Spur in seinem Schreibfeld. Nachdem der Agent eine Linie der Länge  $l_{resource-indicator} \in \mathbb{R}^+$  gemalt hat, geht dieser in den Zustand *Explore* über.

Im *Follow*-Zustand folgt  $a$  einer gefundenen Pheromon-Spur. In jedem Schritt wird überprüft, ob eine Quelle gefunden wurde. Dies geschieht parallel zum Prozess in *Explore*. Falls eine passende Quelle gefunden wurde, wechselt  $a$  in den Zustand *Wait*. Anschließend dreht er sich zum mittels der Funktion  $FIRSTPHEROMONE(resInd)$  gefundenen Punkt auf der Pheromon-Spur. Sobald kein Pheromon mehr gefunden wurde, geht der Agent in *Explore* über.

Insgesamt erkundet die Agenten so die Umgebung, finden zufällig Ressourcen-Quellen, markieren diese und folgen den Markierungen von anderen Agenten.

<sup>13</sup>Dies ist der Fall, wenn der Agent aus *Explore* in *Wait* übergegangen ist. In der algorithmischen Darstellung wird dies negiert durch die Variable  $b_{found-via-indicator}$  repräsentiert.

### 3.5.4 Krankheitsübertragung

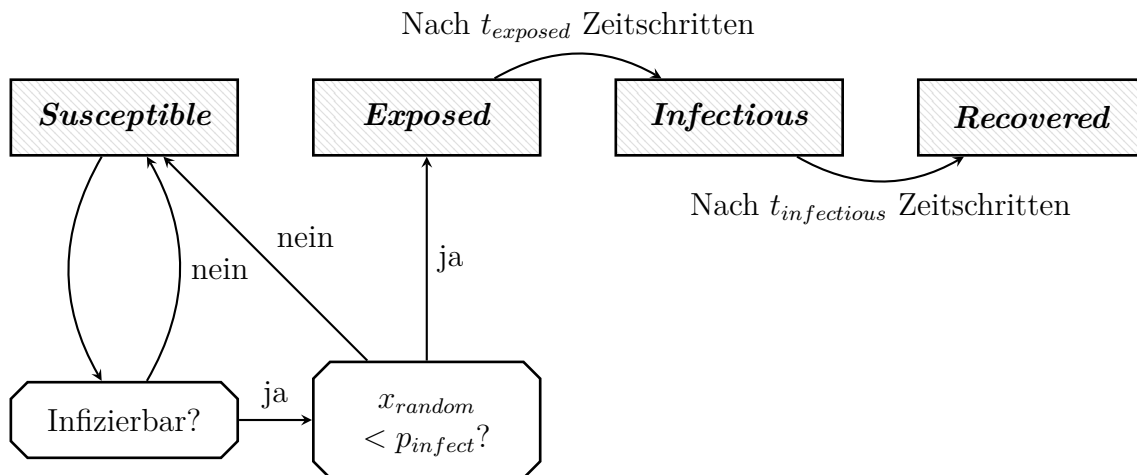
Das Infektionsmodell der Agenten orientiert sich an dem in [Abschnitt 2.1](#) beschriebenen *SEIR*-Modell. Die *SEIR*-Gruppen werden als Zustände der Agenten genutzt. Hierfür ist die Menge der Zustände gegeben als:

$$K = \{Susceptible, Exposed, Infectious, Recovered\} \quad (12)$$

Die Übergänge zwischen den Zustände geschehen strikt linear in der gleichen Reihenfolge wie im Modell. Die Übergangsraten  $\alpha, \beta, \gamma$  wurden durch von der Gesamtpopulation unabhängige Übergangsbedingungen ersetzt. Diese können in jedem Zeitschritt von den Agenten, in Abhängigkeit von Ihrer Umgebung, ausgewertet werden. Das gesamte Übergangsschema ist in [Abbildung 3.6](#) dargestellt.

Die Bedingungen, die  $\alpha$  und  $\gamma$  ersetzen, sind einfache zeitliche Abhängigkeiten. Ein Agent hält sich für  $t_{exposed} \in \mathbb{N}_0$  Zeitschritte im Zustand *Exposed* und für  $t_{infectious} \in \mathbb{N}_0$  Zeitschritte im Zustand *Infectious* auf. Nach dieser Zeit erfolgen die Übergänge in den jeweilig nächsten Zustand.

Der Übergang zwischen den Zuständen *Susceptible* und *Exposed* hat komplexere Bedingungen und stellt die Krankheitsübertragung dar. Hierzu wird überprüft, ob der gegebene Agent infizierbar ist (im folgenden Absatz erklärt) und ist dies der Fall, infiziert sich dieser mit einer gegebenen Wahrscheinlichkeit von  $p_{infect} \in [0, 1]$ . Die Auswertung der Wahrscheinlichkeit erfolgt über den Zufallsparameter  $x_{random}$ .



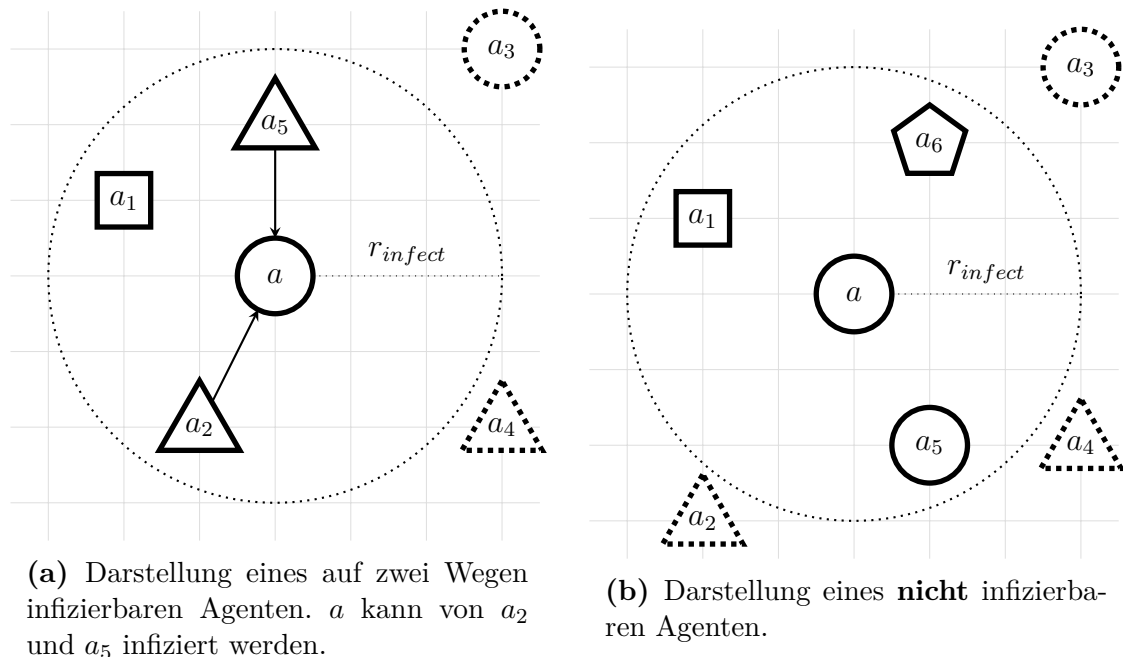
**Abbildung 3.6:** Flussdiagramm zu den Zustandsübergängen der Agenten zwischen den einzelnen *SEIR*-Zuständen.

Für den Agenten  $a$  sei dessen aktueller Zustand hier *Susceptible*.  $a$  wurde noch nicht infiziert und es gilt zu überprüfen, ob  $a$  infizierbar ist. Hierzu sei der Radius  $r_{infect} \in \mathbb{R}^+$  gegeben.  $a$  ist infizierbar, gdw. gilt:

$$\exists b \in A (b \text{ ist im Zustand } Infectious \wedge \text{dist}(a, b) \leq r_{infect}) \quad (13)$$

Diese Bedingung an die Infizierbarkeit ist in [Abbildung 3.7](#) beispielhaft visualisiert.

Sobald  $a$  in den Zustand *Exposed* übergegangen ist, wurde er infiziert. Dieser Infektion ist er sich aber nicht bewusst und kann dementsprechend nicht reagieren. Die Infektion bemerkt er  $t_{realization} \in \mathbb{N}_0$  Zeitschritte nach dem Übergang in den *Infectious*-Zustand. Zum selben Zeitpunkt setzen auch die Symptome der Krankheit ein. Hierzu wird die Bewegungsgeschwindigkeit von  $a$  auf  $v_{agent-symptom} \in \mathbb{R}^+$  gesetzt. So kann eine Verlangsamung der kranken Agenten simuliert werden.



**Abbildung 3.7:** Darstellungen zur Infizierbarkeit von Agenten in der Simulation. Der Infektionsradius  $r_{infect}$  ist um  $a$  eingezeichnet. Alle Agenten in diesem Radius sind mit durchgezogenen Linien umrandet und alle Agenten außerhalb von diesem Radius sind mit gepunkteten Linien umrandet. Die Agenten sind entsprechend ihres *SEIR*-Zustandes als Kreis (*Susceptible*), Quadrat (*Exposed*), Dreieck (*Infectious*) oder Fünfeck (*Recovered*) dargestellt. Potenzielle Infektionen sind als Pfeil vom infizierenden Agenten zum Agent  $a$  dargestellt.

### 3.6 Initial-Zustand des Simulationsmodells

Zu Beginn der Simulation, vor Zeitschritt 0, wird ein leeres Spielfeld nach [Abschnitt 3.2](#) generiert und die notwendigen Modellaspekte werden initialisiert. Im Folgenden wird der für die Reproduzierbarkeit und Vergleichbarkeit der Experimente relevante Prozess der Initialisierung im Detail erläutert.

Alle Agenten werden zufällig im Spielfeld verteilt und orientiert. Hierzu wird die Zufallsvariable  $x_{random}$  genutzt. Zusätzlich benötigen die Agenten *SEIR*-Zustände. Diese sind durch die initialen Größen der *SEIR*-Gruppen  $c_{initial-susceptible}$ ,  $c_{initial-exposed}$ ,  $c_{initial-infectious}$ ,  $c_{initial-recovered} \in \mathbb{N}_0$  gegeben. Insgesamt ist die Initialisierung der Agenten in [Algorithmus 3.5](#) aufgeführt.

Als nächstes werden die Ressourcen-Quellen generiert. Dies geschieht identisch zu den Agenten in einer Schleife und die Quellen werden zufällig auf dem Spielfeld positioniert. Die Anzahl dieser ist durch  $c_{resources} \in \mathbb{N}_0$  gegeben und bleibt für den gesamten Zeitraum der Simulation konstant. Alle Quellen enthalten initial  $s_{resources} \in \mathbb{N}_0$  Ressourcen.

Jedes genutzte Pheromon wird nach dem Verfahren in [Abschnitt 3.4.1](#) konstruiert. Alle Intensitätswerte und die Zeitschritte, in denen diese gesetzt wurden sind mit 0 initialisiert. Dies entspricht einer kompletten Abwesenheit von Pheromonen zu Beginn der Simulation.

Die initialen Positionen, Orientierungen, Zustände, etc. der Modellaspekte werden zusammengenommen als Ausgangszustand des Modells bezeichnet. Alle zufällig berechneten Werte für die Initialisierung werden mittels  $x_{random}$  generiert und vorher wird nicht auf  $x_{random}$  zugegriffen. Die Reihenfolge bei der Generierung von den Positionen, Orientierungen, etc. ist hierbei in verschiedenen Simulationsdurchläufen identisch. Somit resultieren

---

**Algorithmus 3.5** Prozess zur Erstellung und Initialisierung aller Agenten zu Beginn der Simulation.

---

```

1: for  $1 \leq i \leq c_{initial-susceptible}$  do
2:   Erstelle Agent  $a$  in SEIR-Zustand Susceptible.
3:   Generiere mit  $x_{random}$  eine zufällige Position und Rotation für  $a$ .
4: end for
5: for  $1 \leq i \leq c_{initial-exposed}$  do
6:   Erstelle Agent  $a$  in SEIR-Zustand Exposed.
7:   Generiere mit  $x_{random}$  eine zufällige Position und Rotation für  $a$ .
8: end for
9: for  $1 \leq i \leq c_{initial-infectious}$  do
10:  Erstelle Agent  $a$  in SEIR-Zustand Infectious.
11:  Generiere mit  $x_{random}$  eine zufällige Position und Rotation für  $a$ .
12: end for
13: for  $1 \leq i \leq c_{initial-recovered}$  do
14:  Erstelle Agent  $a$  in SEIR-Zustand Recovered.
15:  Generiere mit  $x_{random}$  eine zufällige Position und Rotation für  $a$ .
16: end for

```

---

verschiedene Ausführungen der Simulation mit gleichem *Seed*, gleichen Agenten-Anzahlen und gleicher Ressourcen-Quellen-Anzahl im selben Ausgangszustand. Dies ist wichtig für die Vergleichbarkeit der Experimente in [Abschnitt 4](#). Für alle Experimente werden die selben 1000 *Seeds* und identische Agenten-/Quellen-Anzahlen genutzt. Somit wird jedes Experiment mit den selben 1000 Ausgangszuständen getestet und der Effekt des Zufalls auf die Ergebnisse reduziert. Zusätzlich wird die Reproduzierbarkeit von einzelnen Experimenten sichergestellt.

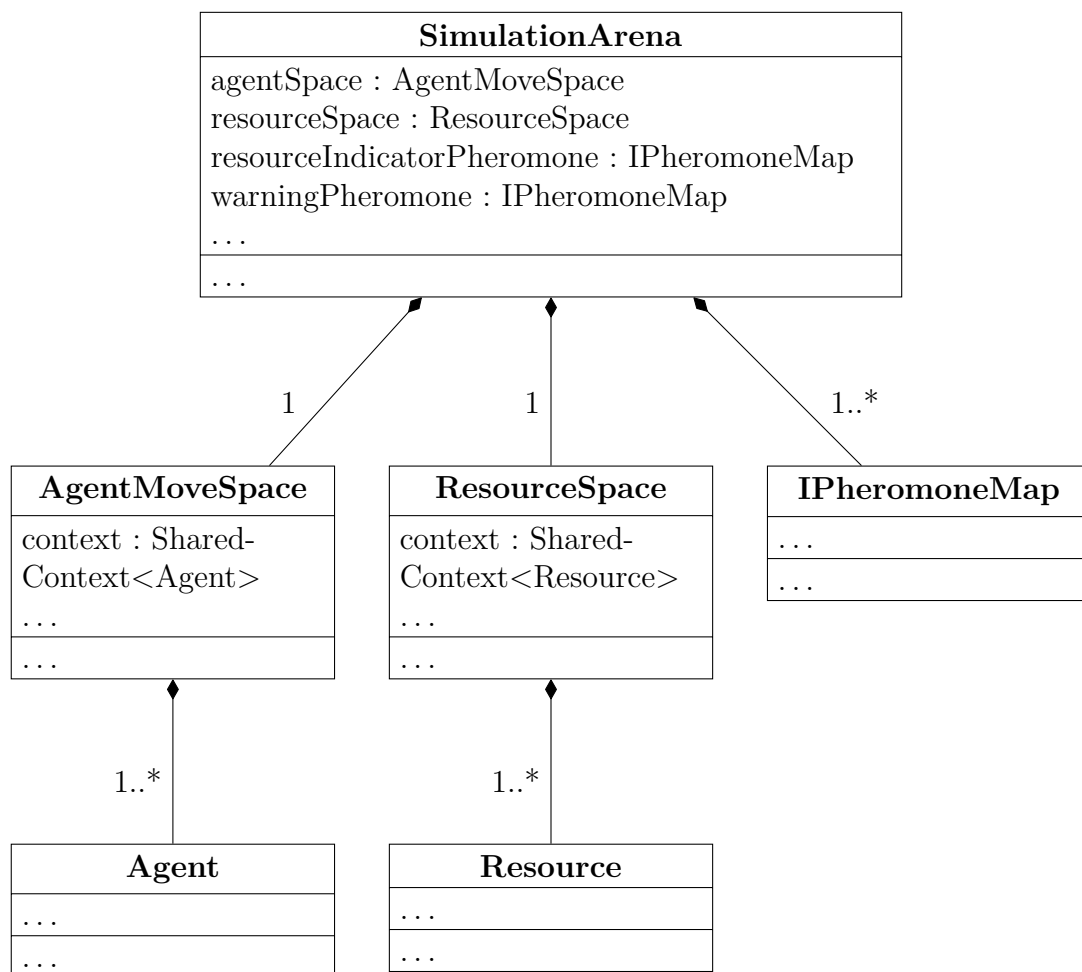
### 3.7 Technische Umsetzung

Als Grundlage für die Umsetzung des Modells wurde die Plattform *Repast-HPC* [48] genutzt. *Repast* (*Recursive Porous Agent Simulation Toolkit*) ist eine *Software*-Bibliothek zur Modellierung von Multi-Agenten-Systemen in verschiedenen Programmiersprachen. Der Hauptableger *Repast-Symphony* [49] ist auf die ursprüngliche *Java*-Implementierung von *Repast* aus dem Jahr 2003 zurückzuführen [50]. Neben diesem Ableger sind über die Jahre Implementierungen in *Python* und *C++* entstanden. An dieser Stelle wird die *C++*-Version genutzt. Diese ist speziell für effiziente parallelisierte Ausführung von komplexen Simulationen ausgelegt. Hierbei nutzt die Plattform das *Message Passing Interface* (*MPI*) [51] zur Kommunikation zwischen parallel laufenden Teilen der Simulation. Diese Version heißt *Repast for High Performance Computing* (*Repast-HPC*).

Die größte Herausforderung der Implementierung mit *Repast-HPC*, war die Synchronisation der verschiedenen Prozesse. Hierzu teilt die Plattform das Spielfeld zunächst in Teilfelder für jeden beteiligten Prozess ein und ermöglicht es Agenten sich zwischen den Prozessen zu bewegen. Über die Einführung von sog. Projektionen, die Agenten im Spielfeld positionieren, werden diese Bewegungen automatisiert. Die Randbereiche der Teilfelder sind von besonderem Interesse, da in diesen Agenten, die von unterschiedlichen Prozessen kontrolliert werden miteinander agieren können (zum Beispiel muss sichergestellt sein, dass Agenten sich über die Teilfeld-Grenzen hinweg gegenseitig infizieren können). In *Repast-HPC* werden hierzu für einen Agenten im Randbereich von Prozess 1 jeweils Kopien dieses Agenten in allen anliegenden Prozessen generiert. Diese „Phantom“-Agenten werden am Ende von je-

dem Zeitschritt mit dem aktuellen Zustand des Agenten synchronisiert<sup>14</sup>. Hierbei gehen alle Änderungen am „Phantom“-Agenten verloren<sup>15</sup>. Diese Projektionen werden in der Simulation für die Agenten und für die Ressourcen-Quellen genutzt. Aus technischer Sicht sind somit die Quellen auch als eigenes Multi-Agenten-System implementiert. Diese Entscheidung wurde getroffen, um keine separate Synchronisation der Ressourcen-Quellen implementieren zu müssen. Für die Pheromone war eine solche Verwendung von vorhandenen Konzepten nicht effizient realisierbar. Diese wurden als zweidimensionales *Array* umgesetzt und manuell am Ende jedes Zeitschritts synchronisiert.

Die verschiedenen Kontexte für die Aspekte der Simulation wurden als einzelne Klassen implementiert und in einer übergeordneten Klasse *SimulationArena* zusammengeführt. Der entstandene Aufbau der Simulationsumgebung ist in [Abbildung 3.8](#) stark vereinfacht<sup>16</sup> dargestellt.



**Abbildung 3.8:** Stark vereinfachte Übersicht des Spielfeldes mit Agenten in der Simulation als *UML*-Diagramm.

<sup>14</sup>Für eine genauere und ansehnlichere Erklärung dieses Problems und der Lösung in *Repast-HPC* können die [offizielle Dokumentation](#) und [Demo 03 aus dem offiziellen Tutorial](#) heran gezogen werden (letzte Öffnungen: 08.09.2023)

<sup>15</sup>Modelle müssen aus diesem Grund anders aufgebaut werden. Ein gutes Beispiel hierfür sind die Krankheitsübertragungen aus [Abschnitt 3.5.4](#). Intuitiver könnten hier die infektiösen Agenten nach Gesunden suchen und diese anstecken, was zu Problemen führt, da Zugriffe auf Agenten im Randgebiet nur die „Phantome“ dieser betrifft und mit der nächsten Synchronisation überschrieben werden.

<sup>16</sup>Für die tatsächliche Implementierung ist eine Vielzahl von Klassen und Methoden zur Synchronisation der einzelnen Aspekte notwendig. Diese und andere Aspekte wurden hier ignoriert.

Für die Ausführung der Simulation muss als erstes *Repast-HPC* installiert werden<sup>17</sup>. Der Quellcode der Simulation kann den [digitalen Anlagen](#) im Ordner */simulation* entnommen werden. Zur Kompilierung kann der bereitgestellte *makefile* genutzt werden. Dieser generiert eine ausführbare Datei, welche mit dem Befehl *mpirun* oder alternativ mit dem *Shell*-Skript *run.sh* ausgeführt werden kann. Zum automatischen Ausführen wurden verschiedene Skripte erstellt, die die Anzahl der Simulationsdurchläufe und die Art der Maßnahmenkonfigurationen steuern.

Die Datei */props/model.props* enthält die Simulationsparameter. Hier sind alle gegebenen Parameter mit ähnlichen Namen zu den in dieser Arbeit genutzten Modellparametern aufgeführt. Die letzten Zeilen der Datei bestimmen, welche Ausgaben für einen Simulationsdurchlauf aufgezeichnet und wo diese gespeichert werden sollen. Standardmäßig wird als Speicherort der Ordner */output* festgelegt.

---

<sup>17</sup>Die notwendigen Dateien und Anweisungen sind [hier](#) zu finden [letzte Öffnung: 04.09.2023]. Verschiedene Abhängigkeiten, wie die *Boost*-Bibliothek und *MPI* müssen vorher installiert werden (genauere Anweisungen sind unter dem gegebenen *link* zu finden).



## 4 Ergebnisse

Die implementierte Simulation wird genutzt, um verschiedene Experimente durchzuführen. Hierzu wird in [Abschnitt 4.1](#) der allgemeine Versuchsaufbau beschrieben. Anschließend wird ein beispielhafter Simulationsdurchlauf zum Infektionsgeschehen vorgestellt ([Abschnitt 4.2](#)). Die ersten Ergebnisse werden in [Abschnitt 4.3](#) präsentiert und geben Vergleichswerte für die restlichen Experimente. Anschließend werden mit Hilfe der Ausgangswerte verschiedene Bewertungskriterien festgelegt. Der Hauptteil der Experimente zu den Eindämmungsstrategien kann in [Abschnitt 4.4](#) gefunden werden. Hier werden auch die zu Grunde liegenden Verhaltensanpassungen eingeführt. In [Abschnitt 4.5](#) sind drei Experimente zu komplexeren Symptomen und Strategien zu finden. Als letztes wurden in [Abschnitt 4.6](#) Anmerkungen zu den Ergebnissen und Problemen bei der Bewertung gesammelt.

Zu allen Experimenten können die gemittelten Verläufe der *SEIR*-Gruppengrößen und der Summe von gesammelten Ressourcen in den [Anlagen 16–23](#) eingesehen werden.

### 4.1 Versuchsaufbau

Für alle Experimente wird eine Grundkonfiguration der Simulation gewählt, die größtenteils unverändert in allen Experimenten beibehalten wird. In den [Anlagen 2–10](#) ist eine Liste aller Werte der Simulationsparameter, jeweils in der Spalte „Standard“ gegeben. In den folgenden Abschnitten werden die wichtigsten Eigenschaften eingeführt.

Die Simulationen werden über  $t_{max} = 2000$  Zeitschritte durchgeführt. Diese obere Grenze wurde experimentell, nach den folgenden Kriterien ermittelt: In dem entstehenden Simulationszeitraum sind für alle Experimente die relevanten Messwerte enthalten und es werden nicht unnötige Zeitschritte, ohne relevante Informationen durchlaufen. Das Spielfeld wird auf eine Größe von  $100 \times 100$  Längeneinheiten gesetzt. Im Spielfeld werden insgesamt drei Ressourcen-Quellen mit einer Größe von zehn verteilt.

Die Pheromon-Zugriffs-Funktionen *resInd* und *warn* werden initialisiert. Die Lebenszeit des Ressourcen-Indikators wird auf  $t_{resource-indicator-lifetime} = 250$  gesetzt. Das Warn-Pheromon wird in [Abschnitt 4.5](#) genutzt, mit einer Lebenszeit von  $t_{warning-lifetime} = 100$ .

Zu Beginn werden insgesamt 100 Agenten in die Arena gesetzt. Hiervon starten 99 im Zustand *Susceptible* und ein Agent in *Exposed*. Zu Beginn sind keine Individuen infizierend oder haben eine Infektion durchlaufen.

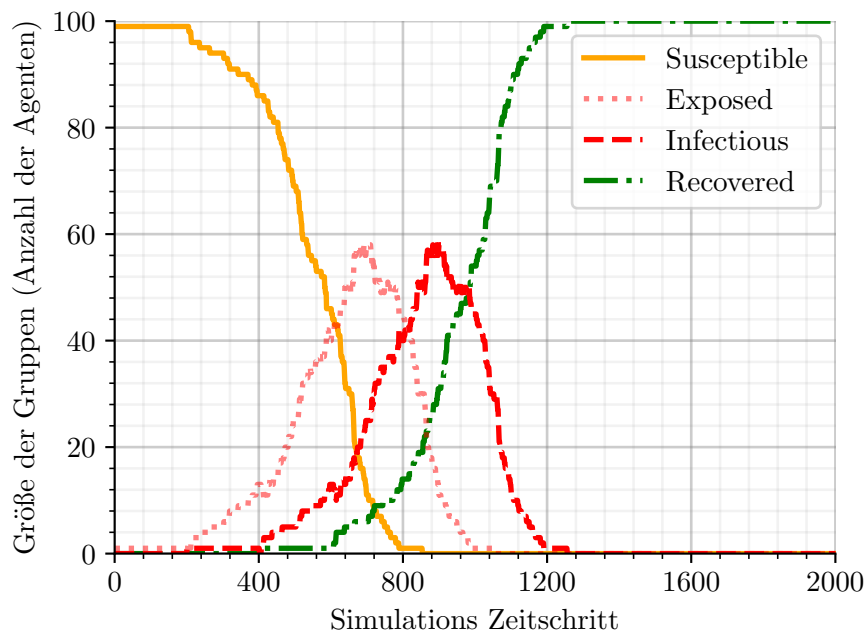
Die Agenten bewegen sich mit einer Geschwindigkeit von einer Längeneinheit pro Zeitschritt über das Spielfeld. Hierbei haben sie in jedem Zeitschritt, in dem sie in einem Umkreis von fünf Längeneinheiten von einem Agenten in *Infectious* sind, eine zehnpromtente Chance sich anzustecken.

Für alle Experimente wird das Verhalten der Agenten angepasst. Anschließend werden mit der gegebenen Konfiguration, aber unterschiedlichen *Random-Seeds* 1000 Simulationen durchgeführt. Die Ergebnisse der einzelnen Simulationen werden zu Experiment-Ergebnissen gemittelt. Für alle Experimente werden die selben *Random-Seeds* genutzt. Hierdurch wird sichergestellt, dass die gleichen Startkonfigurationen getestet werden (siehe [Abschnitt 3.6](#)).



## 4.2 Betrachtung der Krankheitsausbreitung

An dieser Stelle soll die durch das Modell gegebene Krankheitsausbreitung betrachtet werden. Hierzu wird eine Simulation mit der gegebenen Grundkonfiguration ausgeführt. In den [digitalen Anlagen](#) kann hierzu das Video `/model-videos/seir.mp4` betrachtet werden. Dieses stellt die Bewegungen der einzelnen Agenten und deren *SEIR*-Zustand dar. In [Abbildung 4.1](#) sind die Größen der *SEIR*-Gruppen graphisch über die Zeit dargestellt.



**Abbildung 4.1:** Beispiel zum Verlauf des Infektionsgeschehens in der Simulation. Es sind jeweils die Größen der *SEIR*-Gruppen über die Zeit dargestellt.

Die Graphen erscheinen sehr sprunghaft, was auf die diskrete Roboteranzahl zurückzuführen ist, denn es kann nur eine ganzzahlige Anzahl an Robotern in einer *SEIR*-Gruppe sein. Die Graphen wirken verrauscht, was mit den wahrscheinlichkeitsbasierten Infektionen zusammenhängt. Diese beiden Punkte lassen sich durch eine Glättung der Graphen beheben. Für eine ähnliche Simulation wurden die Ergebnisse von 1000 Simulationendurchläufe in jedem Zeitschritt gemittelt (siehe [Anlage 17](#)), was in einen geglätteten Graphen resultiert.

In sowohl den ungeglätteten, als auch den gemittelten Graphen lassen sich Ähnlichkeiten und Unterschiede zu den ursprünglichen, durch Differentialgleichungen ermittelten Graphen des *SEIR*-Modells in [Abbildung 2.2](#) erkennen. Die grundsätzlichen Formen der einzelnen Graphen stimmen überein und die Verläufe der *Susceptible*- und *Recovered*-Gruppen stellen glatte, gegensätzliche S-Kurven dar. Auffällig sind hierbei die zwei abrupten Steigungsänderungen in der *Susceptible*-Kurve<sup>18</sup>, bei den simulierten Ergebnissen. Diese lassen sich durch die Inkubationszeit der Agenten erklären. Zum Beginn der Simulation ist ein Agent in *Exposed* und muss  $t_{exposed}$  Zeitschritte warten, bevor dieser zu *Infectious* wechselt. Ab diesem Zeitpunkt beginnt dieser Agent, andere zu infizieren und die Anzahl der Agenten in *Susceptible* nimmt linear ab. Nach  $2 \cdot t_{exposed}$  Zeitschritten, werden die nächsten Agenten infektiös. Ab diesem Zeitpunkt finden die Übergänge der Agenten kontinuierlich und weniger

<sup>18</sup>Besonders gut sind diese in [Anlage 17](#) zu ca. den Zeitschritten  $t_{exposed} = 200$  und  $2 \cdot t_{exposed} = 400$  zu erkennen.

gebündelt statt. Die Graphen weisen daher keine Unstetigkeiten mehr auf. In den anderen Graphen lassen sich die zwei abrupten Änderungen ebenfalls wiederfinden, da alle restlichen Übergänge zwischen den Infektionszuständen durch feste Anzahlen von Zeitschritten gegeben sind.

Eine weitere Anomalie, die durch die festen Größen entsteht, ist an den Graphen zu den Gruppen *Exposed* und *Infectious* zu erkennen. Diese sind in den Simulationsergebnissen immer, bis auf eine zeitliche Verschiebung, identisch.

### 4.3 Bewertungskriterien und Ausgangswerte

Die Experimente werden auf der Basis von verschiedenen Messwerten bewertet und miteinander verglichen. Es soll jeweils die Eindämmung des Infektionsgeschehen und die Produktivität gemessen werden. Für jeden Simulationsdurchlauf werden die Anzahl der insgesamt gesammelten Ressourcen und die Größen der *SEIR*-Gruppen zu allen Zeitschritten aufgezeichnet. Hierzu werden für jeden Zeitschritt  $t \in T$  folgende Werte gesetzt:

$$\begin{aligned}
 R_t &: \text{Anzahl der bis zum Zeitpunkt } t \text{ eingesammelten Ressourcen} \\
 s_t &: \text{Anzahl der Agenten, die zum Zeitpunkt } t \text{ im Zustand } \textit{Susceptible} \text{ sind} \\
 e_t &: \text{Anzahl der Agenten, die zum Zeitpunkt } t \text{ im Zustand } \textit{Exposed} \text{ sind} \\
 i_t &: \text{Anzahl der Agenten, die zum Zeitpunkt } t \text{ im Zustand } \textit{Infectious} \text{ sind} \\
 r_t &: \text{Anzahl der Agenten, die zum Zeitpunkt } t \text{ im Zustand } \textit{Recovered} \text{ sind}
 \end{aligned}
 \tag{14}$$

Aus diesen Rohdaten können absolute Bewertungskriterien berechnet werden. Für die Produktivität kann hier die Anzahl der insgesamt in der Simulation gesammelten Ressourcen genutzt werden. Dies stellt den maximalen Wert von  $R_t$  dar. Da die Folge monoton steigend ist, ist dies äquivalent zu ihrem finalen Wert  $R_{t_{max}}$ . Somit ergibt sich das Bewertungskriterium  $R_{max}$ :

$$R_{max} = R_{t_{max}} = \max(R_0, \dots, R_{t_{max}}) \tag{15}$$

Ein höherer Wert in diesem Kriterium zeugt von höherer Produktivität.

Für das Infektionsgeschehens ist eine Bewertung komplizierter, da nicht klar ist, was das Ziel einer Eindämmungsstrategie ist. Zum Beispiel könnte hier als einfacher Maßstab die Anzahl der zum Ende der Simulation nie infizierten Agenten  $s_{t_{max}}$  gewählt werden. Dies setzt als Ziel der Eindämmung, die Minimierung der Infektionen insgesamt. In einer Epidemie zeigt dieser Wert nicht die Gefahr einer sich ausbreitenden Krankheit an, da die schwere von Krankheitsverläufen nicht berücksichtigt wird. Bei guten Behandlungsmöglichkeiten sind Infektionen ungefährlicher. Aus diesem Grund soll hier als Ziel der Bewertung die Auslastung des Gesundheitssystem<sup>19</sup> genutzt werden. Diese ist hoch, wenn viele Individuen parallel infiziert sind. Somit kann hier als einfaches Kriterium die in der Simulation maximale Größe der Agenten in *Infectious* oder *Exposed* gewählt werden:

$$I_{max} = \max(i_0, \dots, i_{t_{max}}) (= \max(e_0, \dots, e_{t_{max}}))^{20} \tag{16}$$

Ein niedrigerer Wert für dieses Bewertungskriterium zeugt von einer niedrigeren maximalen Auslastung des Gesundheitssystems, weniger Gefahr durch die Krankheit und somit einer besseren Eindämmung.

<sup>19</sup>Ein Gesundheitssystem existiert im Kontext der Simulation nicht. Dennoch kann hier die Auslastung eines imaginärem Systems wie beschrieben betrachtet werden.

<sup>20</sup>Diese Gleichung gilt für alle Experimente, in denen die Krankheitsausbreitung eingeschaltet wurde. In Experiment 0 ist dies nicht der Fall

Diese Bewertungskriterien werden für alle Simulationsdurchläufe berechnet. Für die Experimente werden anschließend die Verteilungen der Ergebnisse als *Raincloud-Plot* [52] dargestellt (Beispielhaft in [Abbildung 4.2](#) zu sehen). Um die Experimente unterscheiden zu können, wird jedem Experiment eine Identifikationsnummer (*ID*) beginnend mit 0 zugewiesen. Für das Experiment mit der *ID*  $n$  seien  $R_{max}(n)$ ,  $I_{max}(n)$  jeweils die Mediane aller Ergebnisse der 1000 Simulationen zu diesem Experiment.

Die ersten zwei Experimente dienen dem Zweck, Ausgangswerte zu bestimmen, an denen die folgenden Experimente gemessen werden können. Hierbei wird zunächst der Effekt des Infektionsgeschehens auf die Produktivität der Agenten gemessen. Als Experiment 0 wird ein Versuchsdurchlauf gestartet, für den die Krankheitsausbreitung deaktiviert wurde. Alle Agenten bleiben für die gesamte Simulation in ihren ursprünglichen *SEIR*-Zuständen. Somit haben die Symptome der Krankheit keine Auswirkungen auf die Produktivität. Mit der *ID* 1 wird ein Experiment durchgeführt, für das die Krankheitsausbreitung aktiviert wird. Die Ergebnisse dieser Experimente können in [Tabelle 4.1](#) eingesehen werden. Anzumerken ist, dass  $I_{max}(0) = 0$  gilt, da im Experiment 0 kein Agent den Zustand *Infectious* erreicht. Durch die Auswirkungen der Krankheit in Experiment 1, bricht die Produktivität der Agenten um ca. 22% ein

<i>ID</i>	$I_{max}(n)$	$R_{max}(n)$
0	0	135
1	57	106

**Tabelle 4.1:** Die Ergebnisse der Ausgangsexperimente (*ID* 0 und 1) in absoluten Zahlenwerten. Es sind jeweils die Anzahl an durchschnittlich gesammelten Ressourcen und die maximale Anzahl an zu einem Zeitpunkt sich in *Infectious* befindenden Agenten gegeben.

In den folgenden Abschnitten werden verschiedene Änderungen an dem Verhalten der Agenten vorgenommen und deren Auswirkungen mit dem Experiment 1 verglichen. Zu diesem Zweck werden relative Bewertungskriterien definiert. Für ein Experiment mit der *ID*  $n$  sei dessen relative Infektionsmaß als  $I(n)$  und dessen relative Produktivität als  $P(n)$  gegeben:

$$I(n) = \frac{I_{max}(n)}{I_{max}(1)} \quad (17)$$

$$P(n) = \frac{R_{max}(n)}{R_{max}(1)}$$

Basierend auf diesen Wert kann als finales Bewertungskriterium die relative Effizienz  $E(n)$  eines Experiments gegeben werden. Diese setzt die erreichte Eindämmung der Krankheitsausbreitung mit den entstandenen Kosten in der Produktivität in Relation:

$$E(n) = \frac{P(n)}{I(n)} \quad (18)$$

#### 4.4 Strategien zur Eindämmung

Es wurden vier verschiedene Experimente zu simplen Eindämmungsstrategien durchgeführt. Diese werden als Hauptexperimente bezeichnet. Die Experimente ergeben sich aus den Kombinationsmöglichkeiten von den zwei folgenden Verhaltensanpassungen:

1. Agenten im Zustand *Infectious*, die sich ihrer Infektion bewusst sind (/symptomatisch sind) bleiben stehen. Dies kann als eine vereinfachte Form von Isolation angesehen werden, da infektiöse Agenten versuchen, die Krankheiten nicht zu in andere Gegenden des Spielfeldes zu tragen (im Folgenden als *Stehenbleiben* bezeichnet).
2. Agenten im Zustand *Susceptible* können Agenten im Radius von  $r_{infect}$  erkennen und drehen um, falls sie symptomatisch sind. Dies entspricht dem Abstand halten von Individuen, die krank erscheinen (im Folgenden als *Ausweichen* bezeichnet).

Hieraus ergeben sich die vier Kombinationsmöglichkeiten. Die Erste ist das Weglassen beider Anpassungen und somit das bekannte Experiment 1 zu den Ausgangswerten. Anschließend werden in den Experimenten 2 und 3 die Verhaltensanpassungen einzeln getestet. Das Experiment 2 entspricht der ersten Verhaltensanpassung, dem *Stehenbleiben* von Agenten im Status *Infectious* und das Experiment 3 entspricht dem *Ausweichen* für Agenten in *Susceptible*. Im Experiment 4 werden beide Anpassungen zusammen getestet.

Die Verteilung der Ergebnisse von allen Simulationsdurchläufe für die Hauptexperimente sind zusammen mit den Ausgangswerten in [Abbildungen 4.2](#) und [4.3](#) dargestellt. Es ist anzumerken, dass für Experiment 0 die Krankheitsausbreitung deaktiviert war und somit alle Werte von  $I_{max}$  immer 0 sind. Für alle sonstigen Ergebnisse können die Median-Werte des Experimentes an den *Box-Plots* abgelesen werden. Die exakten Werte sind, zusammen mit den errechneten relativen Ergebnissen, in [Tabelle 4.2](#) gegeben.

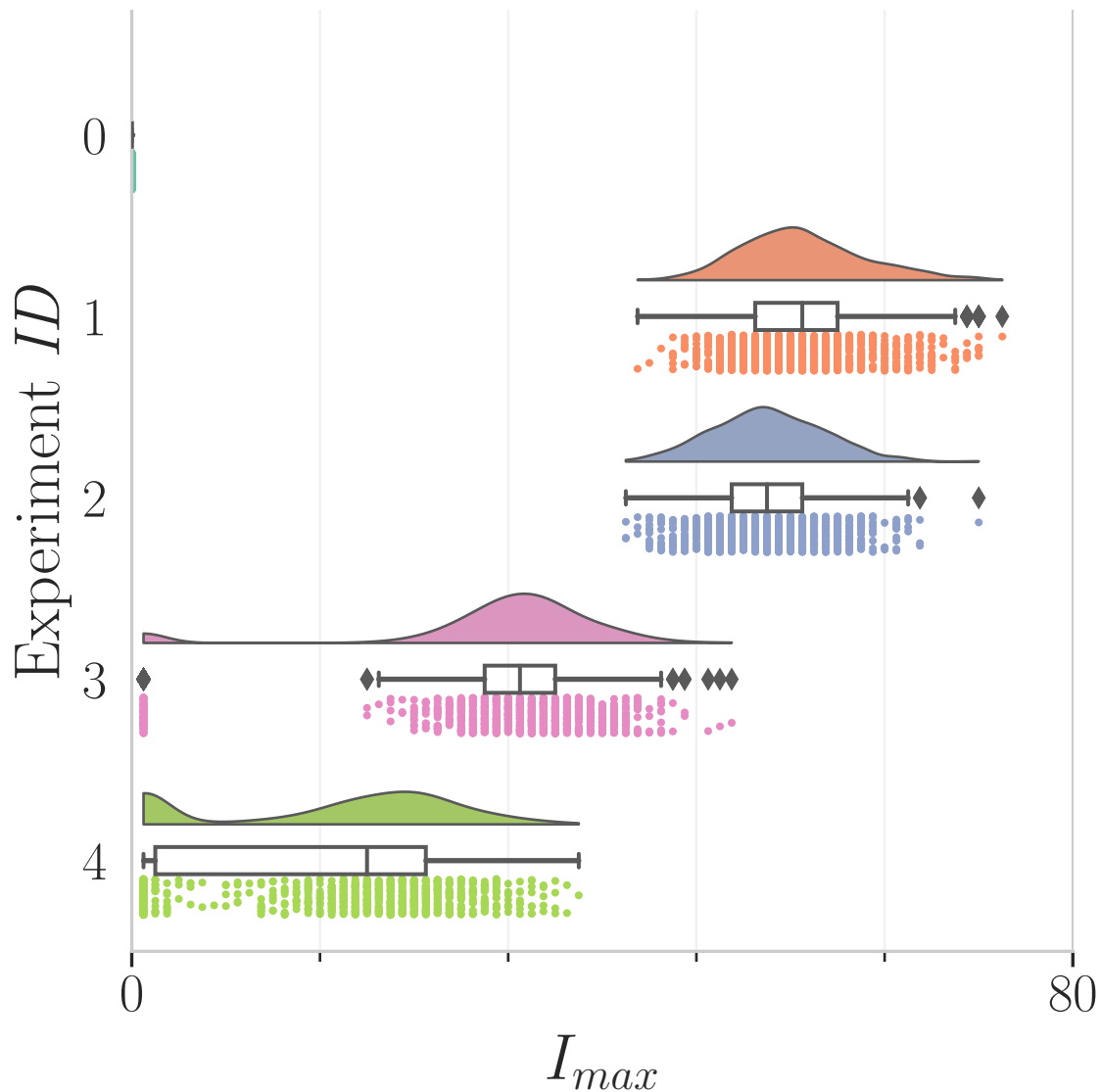
<i>ID</i>	$I_{max}$ (n)	$R_{max}$ (n)	<b>I</b> (n)	<b>P</b> (n)	<b>E</b> (n)
1	57	106	1,00	1,00	1,00
2	54	102	0,95	0,96	1,02
3	33	95	0,58	0,90	1,55
4	20	96	0,35	0,91	2,58

**Tabelle 4.2:** Ergebnisse der Hauptexperimente. Es sind sowohl die absoluten Mediane der einzelnen Simulationen, als auch die relativen Ergebnisse zu Experiment 1 gegeben.

An den Ergebnissen sind interessante Veränderungen der Infektionszahlen zu erkennen. Das Experiment 2 zeigt hier keine große Eindämmung der Krankheitsausbreitung mit einem Infektionsmaß von 95%. Das *Stehenbleiben* von Agenten in *Infectious* alleine ist somit in dieser Hinsicht keine effektive Verhaltensanpassung. In Experiment 3 ist eine größere Veränderung zu beobachten. Hier fällt  $I(3)$  auf 58%. Auffällig ist, dass obwohl der Effekt in Experiment 2 nicht besonders groß war, die Hinzunahme der Anpassung *Stehenbleibens* zum *Ausweichen* in Experiment 4 einen wesentlich größeren Einfluss hat. Hier fällt  $I(4)$  im Vergleich zu  $I(3)$  um 23% auf 35%. Anzumerken ist weiterhin, dass in Simulationsdurchläufen von den Experimenten 3 und 4 die Ausbreitung der Krankheit teilweise komplett verhindert wird. In Experiment 4 macht dies ca. 24% aller Simulationen aus.

Die Veränderung der Produktivitätsmaße fällt deutlich geringer als die Veränderung des Infektionsmaßes aus<sup>21</sup>. Auch hier ist ein schrittweiser Abfall zu erkennen. In Experiment 2 fällt die Produktivität auf 96%. Die Experimente 3 und 4 liegen hier tiefer, bei 90% und 91%, wobei der Unterschied zwischen diesen Werten unter Betrachtung der Stichprobengröße nicht ausschlaggebend ist.

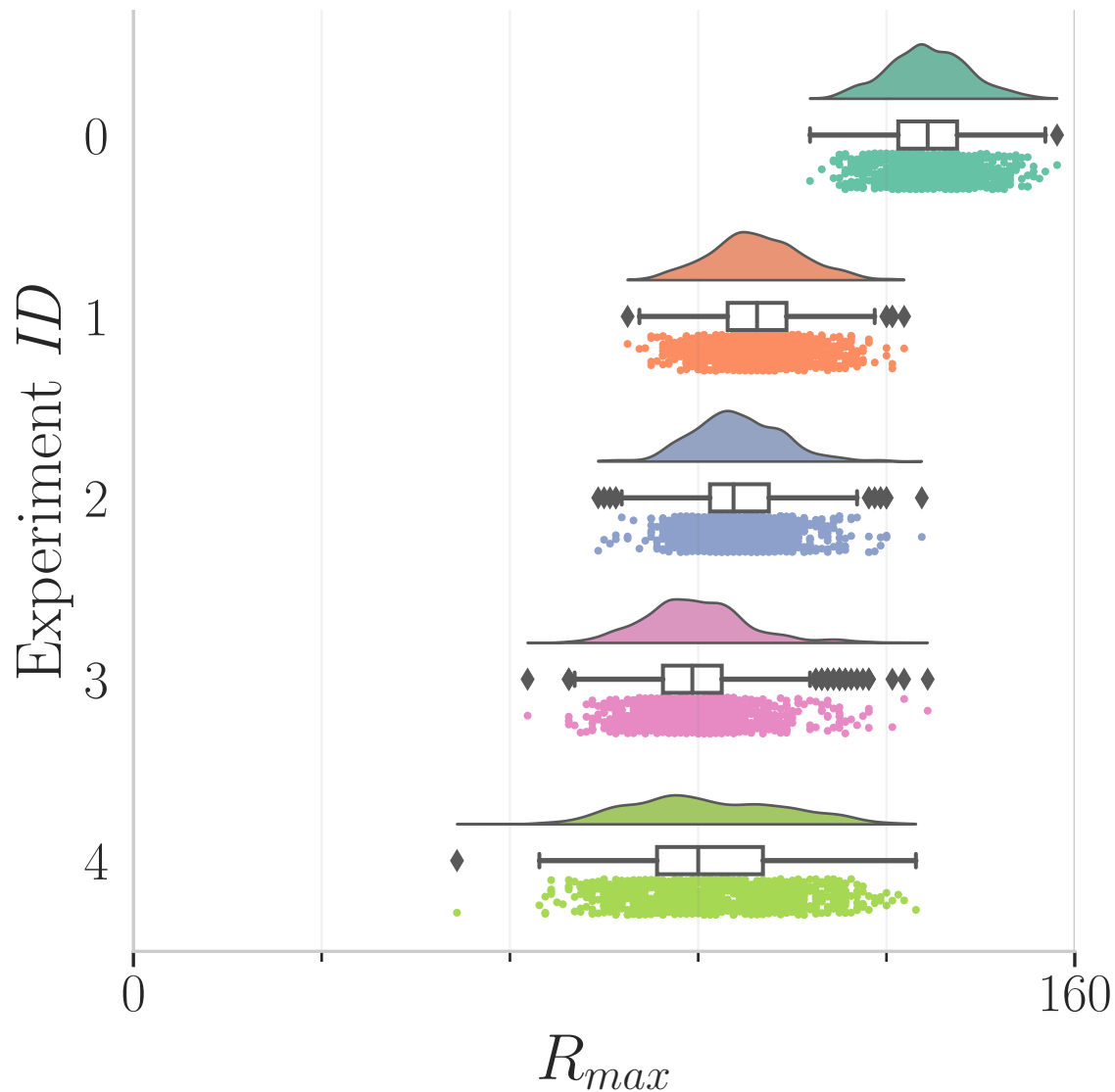
<sup>21</sup>Dies hängt eventuell mit der Messweise zusammen. Siehe [Abschnitt 4.6](#)



**Abbildung 4.2:** Verteilung der Werte von  $I_{max}$  für alle Simulationsdurchläufe der vier Hauptexperimente und der Ausgangsexperimente.

Auf Grund von den ähnlichen Produktivitätsmaßen, spiegeln die Effizienz-Werte ein ähnliches Bild zu den Infektionsmaßen wieder. Durch die Definition des Effizienzmaßes werden die Unterschiede zwischen den Experimente größer, da sie nicht mehr linear dargestellt werden. Experiment 2 erreicht mit einem Wert von 1,02 eine kaum bessere Effizienz als das Ausgangsexperiment 1, da die geringe Infektionseindämmung von ähnlichem Niveau zum Produktivitätseinbruch ist. Mit Experiment 3 wird das Effizienzmaß interessanter. Hier findet eine deutlich stärkere Eindämmung statt und die Produktivität sinkt nur unerheblich weiter. Das Resultat ist eine Effizienz von 1,55. In Experiment 4 steigt die Effizienz auf den höchsten gemessenen Wert von 2,58.

Insgesamt ist zu beobachten, dass die Verhaltensanpassung des *Ausweichens* der Agenten in *Susceptible* eine sehr effektive und effiziente Maßnahme zur Eindämmung des Infektionsgeschehens zu sein scheint. Entgegen hierzu liefert das *Stehenbleiben* von infektiösen Agenten alleine keine guten Ergebnisse. Obwohl die infektiöse Agenten *stehenbleiben*, kommen andere Individuen ihnen nahe, da die Agenten in *Susceptible* sich frei bewegen. So wird die



**Abbildung 4.3:** Verteilung der Werte von  $R_{max}$  für alle Simulationsdurchläufe der vier Hauptexperimente und der Ausgangsexperimente.

Krankheit stark verbreitet. In Experiment 4 ist interessanterweise zu beobachten, dass das *Stehenbleiben* einen großen Effekt hat, wenn es in Verbindung mit dem *Ausweichen* der Agenten in *Susceptible* verbunden wird. Dies lässt sich dadurch erklären, dass es einfacher ist, stationären Gefahren auszuweichen, als solchen, die sich bewegen.

## 4.5 Weitere Experimente

Neben den Hauptexperimenten, wurden drei zusätzliche Experimente durchgeführt, dessen Ziel es ist, verschiedene interessante Verhaltensanpassung auszutesten. Das Ziel dieser Experimente ist es nicht, die Sachverhalte ausführlich zu erforschen, sondern die Implementierbarkeit und Analyse von komplexeren Verhaltensmustern beispielhaft zu beobachten.

Das erste dieser Experimente (*ID*: 5) beschäftigt sich mit der *Gruppierung* (engl. *Clustering*) von infektiösen Agenten. Hierzu wird das in eingeführte Warnungs-Pheromon genutzt. Sobald ein Agent symptomatisch wird und sich somit seiner Krankheit bewusst ist, beginnt dieser eine Pheromon-Spur hinter sich zu legen. Andere Agenten im gleichen Zustand fol-

gen dieser Pheromon-Spur und legen parallel Pheromone. Alle anderen Agenten ignorieren die gelegten Spuren. Das Ziel hierbei ist es, die infektiösen Agenten zu *Gruppieren* und so die räumliche Ausbreitung der Krankheit einzuschränken, ohne die Produktivität der unbetroffenen Agenten zu beeinträchtigen.

Die Wahrnehmungen von Pheromonen ist für die Ressourcen-Suche der Agenten ein zentraler Aspekt und kann mit einer Art von Riechen bei Menschen verglichen werden. Während der *SARS-CoV-2*-Pandemie waren Einschränkungen des Geruchssinn eines der wichtigsten Symptome einer Infektion [53]. Aus diesen Gründen beschäftigen sich die Experimente 6 und 7 mit der Beeinträchtigung der Pheromon-Wahrnehmung von symptomatischen Agenten. In Experiment 6 wird für alle diese Agenten die Wahrnehmung von Pheromonen komplett deaktiviert und sie können somit keinen Pheromon-Spuren mehr folgen. Das Experiment 7 baut hierauf auf und lässt symptomatische Agenten zufällig Phantom-Pheromon-Spuren<sup>22</sup> erkennen und diesen folgen.

Die Experimente wurden parallel zu den Hauptexperimenten durchgeführt und aufgearbeitet. In [Abbildungen 4.4](#) und [4.5](#) sind die Verteilungen der gemessenen absoluten Ergebnisse, zusammen mit den Ausgangswerten zu sehen. Die gemittelten Experiment-Ergebnisse und die relativen Bewertungskriterien sind in [Tabelle 4.3](#) gegeben.

<i>ID</i>	$I_{\max}(\mathbf{n})$	$R_{\max}(\mathbf{n})$	$I(\mathbf{n})$	$P(\mathbf{n})$	$E(\mathbf{n})$
1	57	106	1,00	1,00	1,00
5	55	108	0,96	1,02	1,06
6	56	101	0,98	0,95	0,97
7	56	102	0,98	0,96	0,98

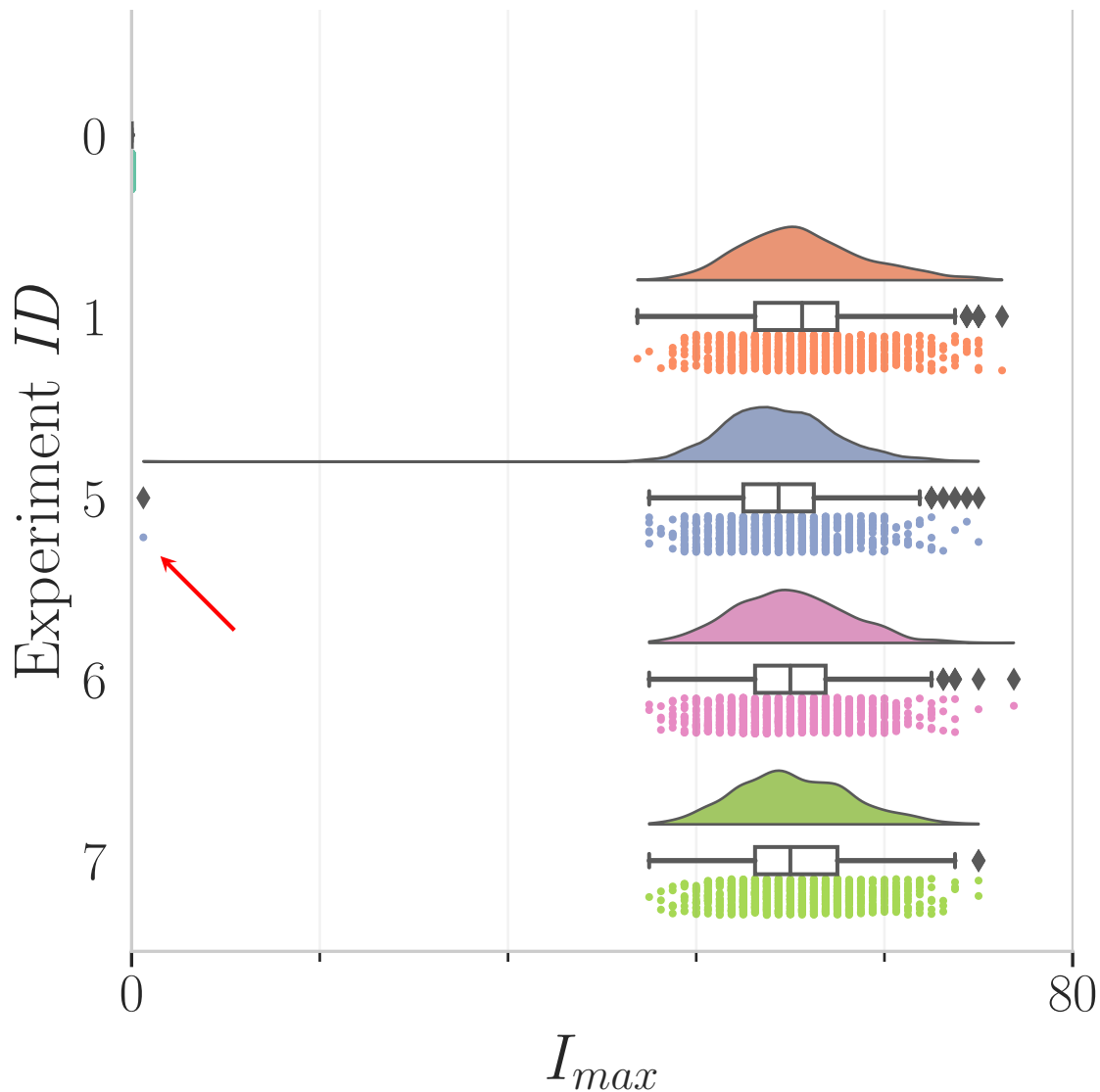
**Tabelle 4.3:** Ergebnisse der Zusatzexperimente. Es sind sowohl die absoluten Mediane der einzelnen Simulationen, als auch die relativen Ergebnisse zu Experiment 1 gegeben.

Die Ergebnisse der Zusatzexperimente fallen, im Vergleich zu den Hauptexperimenten, deutlich weniger ausschlaggebend aus. Hierbei sinkt das Infektionsmaß in Experiment 5 auf 96%. Interessanterweise schafft es eine Simulation, die Krankheitsausbreitung vollständig einzudämmen. Dies ist an dem einem Extremwert in [Abbildung 4.4](#) zu erkennen. Der zum Beginn des Durchlaufes infizierte Agent schafft es keine Agenten zu infizieren. Zurückzuführen ist dies auf die Zufälligkeit der Bewegungen und Infektionen von Agenten. Für die Experimente 6 und 7 sinkt  $I(n)$  auf 98%. Es ist anzumerken, dass eine wirkliche Eindämmung des Infektionsgeschehens für Experimente 6 und 7 nicht zu erwarten war, da die hier getesteten Verhaltensanpassungen keine geplanten Auswirkungen auf die Krankheitsausbreitung haben.

Bei den Produktivitätsmaßen ist dies anders. Hier haben die Experimente zur Veränderung der Wahrnehmung explizit angesetzt. In den Ergebnissen ist ein geringer Einbruch zu erkennen.  $P(n)$  fällt auf 95% für Experiment 6 und 96% in Experiment 7. Dieser Rückgang kann in den Verteilungen wiedererkannt werden und fällt geringer aus als erwartet. In Experiment 5 steigt die Produktivität.  $P(5)$  liegt bei 102%. Diese Erhöhung ist nicht stark.

<sup>22</sup>Mit diesem Begriff sind Pheromon-Spuren gemeint, die in dem Modell nicht existieren. Ein Agent nimmt Pheromone wahr, die nicht vorhanden sind. Auf einer technischen Ebene müssen für die Simulation keine falschen Pheromon-Spuren eingeführt werden. Stattdessen beginnen betroffene Agenten in jedem Zeitschritt mit einer Wahrscheinlichkeit von  $p_{\text{phantom-start}} \in [0, 1]$  geradeaus zu laufen und hören dann mit einer Chance von  $p_{\text{phantom-stop}} \in [0, 1]$  auf.



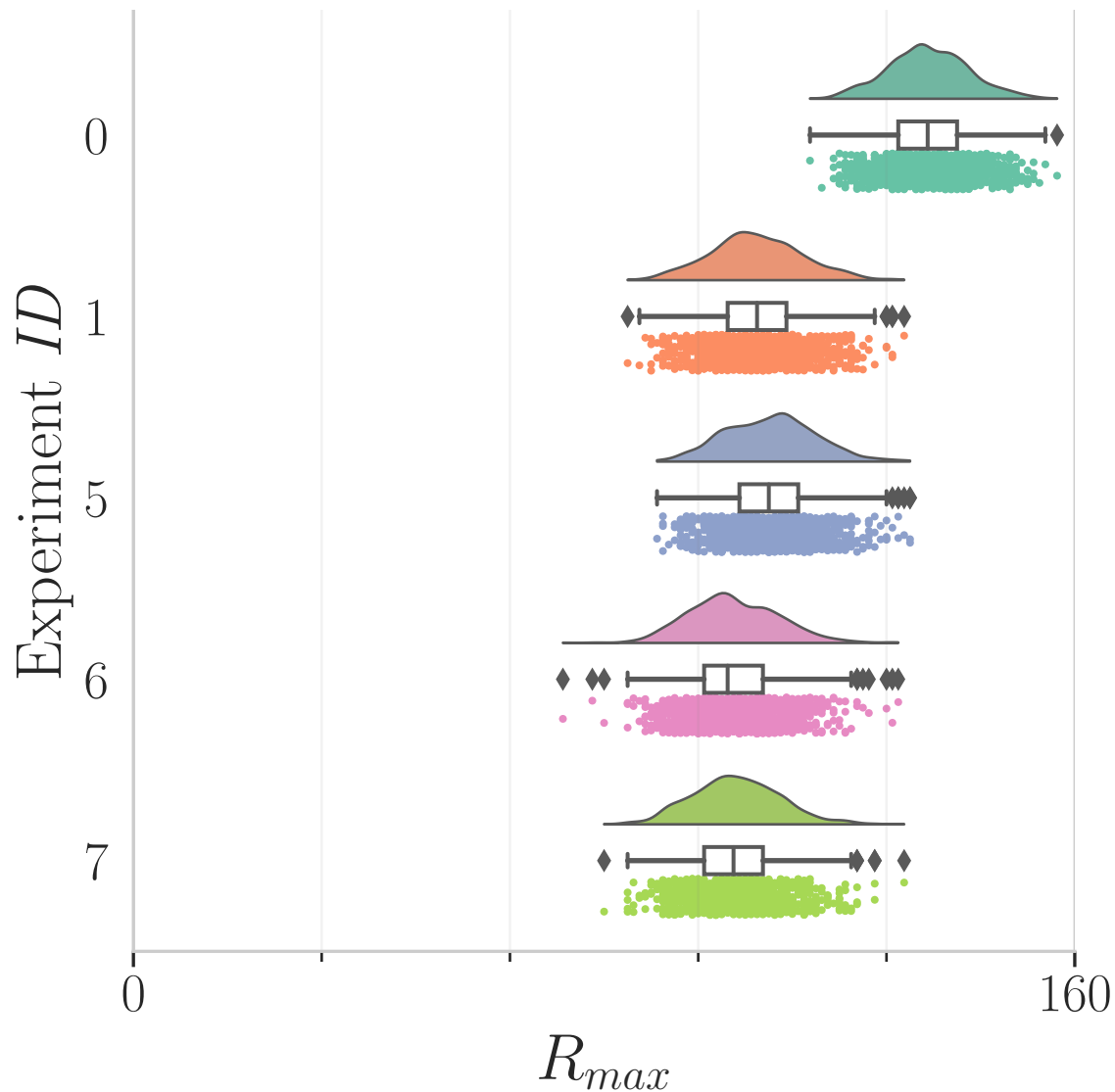


**Abbildung 4.4:** Verteilung der Werte von  $I_{max}$  für alle Simulationsdurchläufe der drei Zusatzexperimente und der Ausgangsexperimente. Der rote Pfeil markiert einen auffälligen Extremwert, der im Text angesprochen wird.

Die Effizienzen verändern sich bei den Zusatzexperimenten nur geringfügig und spiegeln die genannten Veränderungen des Infektionsmaßes und der Produktivität wieder.

Zusammenfassend lässt sich sagen, dass die *Gruppierung* von Agenten geringfügige, rein positive Ergebnisse ohne negative Effekte zu erzielen scheint und für weitere Untersuchungen interessant sein könnte. In den Experimenten zur Veränderung der Pheromon-Wahrnehmung wurden keine ausschlaggebenden Ergebnisse erzielt, die klare Schlussfolgerungen zulassen. Im allgemeinen sind die Ergebnisse mit den gewählten Simulationsparametern nicht sehr aussagekräftig





**Abbildung 4.5:** Verteilung der Werte von  $R_{max}$  für alle Simulationsdurchläufe der drei Zusatzexperimente und der Ausgangsexperimente.

## 4.6 Anmerkungen

In den Ergebnissen ist auffällig, wie geringfügig sich die Produktivität zwischen den einzelnen Experimenten verändert. Insgesamt fällt  $P(n)$  im Minimum auf 90%, was den Schluss zulässt, dass im Allgemeinen der wirtschaftliche Schaden in den Experimenten nicht groß ist. Allerdings wird hierbei nicht beachtet, wie groß der Produktivitätseinbruch maximal sein kann. Anhand von zwei Feststellungen lässt sich zeigen, dass ein theoretische Maximalwert wesentlich tiefer als bei 100% liegt:

1. Die getesteten Verhaltensanpassungen betreffen alle nur die symptomatischen Agenten (*Stehenbleiben* und *Gruppierung*) oder stellen eine Reaktion auf symptomatische Agenten dar (*Ausweichen*). Somit verändert sich das Verhalten von Agenten nur während infektiöse Agenten existieren und der Effekt auf die Produktivität ist zeitlich begrenzt. Bei der Betrachtung der *SEIR*-Graphen, wie [Anlage 17](#) wird klar, dass dieser Zeitraum nur einen Teil der gesamten Simulation ausmacht (in diesem Fall ca. vom Zeitschritt 200 bis zum Zeitschritt 1400). Für den Rest der Zeit ist die Produktivität identisch zum Ausgangsexperiment 1.

2. Aus der selben Argumentation ergibt sich auch, dass zu jedem Zeitpunkt nur eine Subgruppe der Gesamtpopulation in ihrer Produktivität eingeschränkt ist. Die Größe dieser ist abhängig von der Anzahl an symptomatischen Agenten zu diesem Zeitpunkt und somit der Höhe des *Infectious*-Graphen.

Zusammengenommen lässt sich sagen, dass der maximale Produktivitätseinbruch proportional zur prozentualen Fläche unter dem *Infectious*-Graph in den *SEIR*-Graphen ist. Diese liegt bei ca. 10% und ist somit deutlich eingeschränkt. Es ist anzumerken, dass dies nicht direkt auf einen maximalen Produktivitätseinbruch von 10% schließen lässt, aber zeigt, dass dieser begrenzt ist.

Neben der Betrachtung der Krankheitsausbreitung in [Abschnitt 4.2](#), sollte die Ressourcen-Suche der Agenten getestet werden. Hierzu wurde ein Experiment durchgeführt, in dem die Agenten anstelle der gegebenen Ressourcen-Suche eine reine Exploration verwenden. Somit bewegen sich die Agenten zufällig in dem Spielfeld und suchen Ressourcen-Quellen. Sie helfen sich nicht gegenseitig mit Pheromon-Spuren. Im Vergleich zwischen diesem und Experiment 0 lässt sich die Veränderung der Produktivität und somit die Effektivität der Ressourcen-Suche messen. In dem neuen Experiment lag  $R_{max}$  bei 159. Somit sorgt die Nutzung von Pheromon-Spuren für eine Produktivitätsreduktion von ca. 15%<sup>23</sup>. Die Nutzung der Ressourcen-Suche ist für die gegebenen Simulationsparameter nicht sinnvoll. Durch die Tatsache, dass die Bewegungsgeschwindigkeit eine entscheidende Rolle in der Produktivität spielt und den Vergleich zwischen Experimenten, in denen die Ressourcen-Suche genutzt wird, verfälscht dies aber nicht die Ergebnisse.

Eine Wiederholung der Experimente und Auswertung mit angepassten Methoden kann aus zeitlichen Gründen an dieser Stelle nicht durchgeführt werden. Eine Betrachtung der genannten Probleme sind für Folgearbeiten interessant (siehe [Abschnitt 5.2](#)).

---

<sup>23</sup>In den ersten Ergebnissen der Simulation wurden leicht veränderte Parameter verwendet und die Ressourcen-Suche hatte positive Auswirkungen auf die Produktivität. Dieses Experiment zur Effektivität der Ressourcen-Suche wurde spät mit den hier gegebenen Ergebnissen durchgeführt und eine vollständige Wiederholung der Experimente kam nicht mehr in Frage.

## 5 Zusammenfassung

Die vorliegende Arbeit untersuchte, wie Strategien zur Eindämmung von Infektionskrankheiten mithilfe von Multi-Agenten-Systemen simuliert und analysiert werden können. Die Ausbreitung von Krankheiten wurde anhand des *SEIR*-Modells modelliert. Zur Messung der Produktivität der Gesamtpopulation wurden Ressourcen eingeführt, die von den Agenten eingesammelt werden. Dies ermöglicht die Betrachtung des wirtschaftlichen Schaden durch den Produktivitätsverlust, wie auch die Vermeidung desselben. Die Agenten kommunizieren untereinander über Pheromon-Spuren um Ressourcen aufzuspüren. Das gesamte Modell wurde als *Software*-Simulation mithilfe der Plattform *Repast-HPC* implementiert.

Mittels der entstandenen Simulation wurden die Auswirkungen von zwei verschiedenen Verhaltensanpassungen und deren Kombination untersucht. Weiterhin fand eine beispielhafte Betrachtung von komplexeren Eindämmungsstrategien und Symptomatiken statt. In den Experimenten wurde der Effekt der Strategien auf die Eindämmung der Krankheit und auf die Veränderung der Produktivität absolut und in Relation zum Ausgangswert bewertet.

### 5.1 Fazit

Die Analyse der Eindämmungsstrategien hat gezeigt, dass die Vermeidung von Kontakt zu symptomatischen Individuen effektiv Infektionen verhindert. Symptomatische Agenten können hierbei die Effektivität erhöhen, indem sie ihre Bewegungen reduzieren, wobei diese Anpassung alleine nicht effektiv ist. Der Effekt auf die Produktivität blieb hierbei gering. Die komplexeren Symptome, wie die Einschränkung der Pheromon-Wahrnehmung, und die komplexere Verhaltensanpassung, das *Gruppieren* der infektiösen Agenten durch Pheromon-Spuren, hatten keinen großen Effekt auf die Krankheitsausbreitung oder die Produktivität.

Insgesamt lässt sich festhalten, dass die Effektivität und Effizienz von Eindämmungsstrategien stark von den Umgebungsfaktoren (hier Simulationsparameter) abhängt. Dies zeigt sich besonders an den Anmerkungen in [Abschnitt 4.6](#). Es ist deutlich geworden, dass die Kombination von verschiedenen Strategien für die effektive Krankheitsbekämpfung von essenzieller Bedeutung ist.

### 5.2 Ausblick

Während der Bearbeitung der Thematik und speziell den Experimenten sind verschiedene interessante Aspekte aufgefallen, die im begrenzten zeitlichen Rahmen dieser Bachelorarbeit nicht thematisiert werden konnten. Hier soll eine Übersicht von Ansatzpunkten für Folgearbeiten gegeben werden.

Wie erwähnt, haben die gewählten Simulationsparameter einen massiven Einfluss auf die Ergebnisse. Es scheinen verschiedene Parameter größere Auswirkungen zu haben als andere, wie es sich in kleineren Proben zur Festsetzung der Parameterwerte gezeigt hat. Zum Beispiel wäre eine Untersuchung der folgenden Aspekte interessant:

1. **Inkubationszeit:** Die Zeit, die ein Agent im *Exposed*-Zustand verbringt, kann von entscheidender Bedeutung für die räumliche Ausbreitung der Krankheit sein, da dieser noch nicht weiß, dass er infiziert ist und die Krankheit unbewusst in andere Bereiche des Spielfeldes trägt.

2. **Realisationszeit:** Anschließend an die Inkubationszeit wechselt ein Agent in den Zustand *Infectious* und kann andere Agenten anstecken. Allerdings realisiert der infektiöse Agent dies erst nach  $t_{realization}$  Zeitschritten und der einsetzenden Symptomatik. Somit reagiert der Agent um diese Zeit verzögert und kann erst anschließend andere Agenten schützen. Eine ähnliche Verzögerung konnte bei *SARS-CoV-2*-Infektionen beobachtet werden [54].

Eine grundsätzliche Verbesserung der Ergebnisauswertung könnte präzisere Ergebnisse hervorbringen. Die Problematik der erhobenen Werte, speziell zur Produktivität wurde in [Abschnitt 4.6](#) aufgeführt. Hier kann angesetzt und bei der Bewertung die gegebenen Einschränkungen des maximalen Produktivitätseinbruches beachtet werden.

Eine der sinnvollsten Folgeuntersuchungen stellt die Betrachtung von weiteren Eindämmungsstrategien dar. Hier gibt es verschiedene Ansätze zur Wahl der Strategien. Ein Ansatz wäre die Orientierung an in der echten Welt genutzten Maßnahmen. Es könnte zum Beispiel eine Form der Quarantäne von symptomatischen Agenten oder auch ein kompletter *Lockdown* zur vollständigen Unterbindung der Krankheitsausbreitung untersucht werden. Interessant ist auch die Betrachtung der Auswirkungen von Impfungen. Hierzu kann das genutzte Infektionsmodell um eine Gruppe von geimpften Agenten, analog zum *SIRV*-Modell [55] erweitert werden. Neben diesen Ansätzen kann sich am Simulationsmodell orientiert werden. Zum Beispiel sollte das Warn-Pheromon ursprünglich von infektiösen Agenten verteilt werden, um infizierbare Individuen zu warnen. Das direkte Ausweichen in Experiment 3 und 4 ersetzt dieses Konzept. Dennoch ist die Untersuchung einer indirekten Variante über Pheromone für Applikationen interessant, in denen Agenten andere Individuen in ihrer Umgebung nicht direkt erkennen und als infektiös identifizieren können<sup>24</sup>.

Für komplexere Strategien muss das Modell erweitert werden, zum Beispiel bei der Betrachtung von Impfungen. Neben dieser Erweiterung ist auch der allgemeine Ausbau interessant. Hier kann zum Beispiel auf die Akzeptanz der Agenten verschiedener Eindämmungsstrategien eingegangen werden. In der echten Welt hat sich gezeigt, dass die Bevölkerung nicht zu 100% alle verordneten Maßnahmen annimmt und kooperiert. Der Effekt von Agenten, die sich gegenüber bestimmten Strategien verweigern, wäre hier zu untersuchen. Zusätzlich kann der Gedanke weitergeführt und untersucht werden, wenn Agenten sich nicht bloß verweigern, sondern aktiv gegen Maßnahmen vorgehen. Dies kann als Böswilligkeit aufgefasst werden.

Zuletzt sei hier die Umsetzung des Modells als Echtwelt-Simulation genannt, welches durch die Anlehnung an die DeziBots von Anfang an eine Idee für Folgearbeiten war. Hierbei wäre es interessant zu sehen, wie die Hinzunahme der realistischen Umgebungsinteraktionen in einer Echtwelt-Simulation die Ergebnisse gegenüber der *Software*-Simulation verändert. Allerdings erfordert diese Umsetzung einen erheblichen Aufwand in der Implementierung.

---

<sup>24</sup>Dies stellt einen Großteil der Anwendungen dar. Die in den Experimenten gewählte Form der direkten Erkennung ist sehr mächtig, aber schwer umzusetzen.

## Literaturverzeichnis

- [1] RKI, „Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019 (COVID-19),“ Robert Koch-Institut, Robert Koch-Institut, Nordufer 20, 13353 Berlin, Tech. Rep., 03 2020. [Online]. Verfügbar: [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Situationsberichte/2020-03-11-de.pdf?\\_\\_blob=publicationFile](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/2020-03-11-de.pdf?__blob=publicationFile)
- [2] H. Yang und J. Ma, „How an Epidemic Outbreak Impacts Happiness: Factors that Worsen (vs. Protect) Emotional Well-being during the Coronavirus Pandemic,“ *Psychiatry Research*, Vol. 289, S. 113045, 2020. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0165178120309902>
- [3] M. Richards, M. Anderson, P. Carter, B. L. Ebert, und E. Mossialos, „The impact of the COVID-19 pandemic on cancer care,“ *Nature Cancer*, Vol. 1, Nr. 6, S. 565–567, Jun 2020. [Online]. Verfügbar: <https://doi.org/10.1038/s43018-020-0074-y>
- [4] A. Pak, O. A. Adegboye, A. I. Adekunle, K. M. Rahman, E. S. McBryde, und D. P. Eisen, „Economic Consequences of the COVID-19 Outbreak: the Need for Epidemic Preparedness,“ *Frontiers in Public Health*, Vol. 8, 2020. [Online]. Verfügbar: <https://www.frontiersin.org/articles/10.3389/fpubh.2020.00241>
- [5] W. O. Kermack, A. G. McKendrick, und G. T. Walker, „A contribution to the mathematical theory of epidemics,“ *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 115, Nr. 772, S. 700–721, 1927. [Online]. Verfügbar: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1927.0118>
- [6] M. J. Wonham, T. de Camino-Beck, und M. A. Lewis, „An epidemiological model for West Nile virus: invasion analysis and control applications,“ *Proceedings of the Royal Society of London. Series B: Biological Sciences*, Vol. 271, Nr. 1538, S. 501–507, 2004. [Online]. Verfügbar: <https://royalsocietypublishing.org/doi/abs/10.1098/rspb.2003.2608>
- [7] Y.-C. Chen, P.-E. Lu, C.-S. Chang, und T.-H. Liu, „A Time-Dependent SIR Model for COVID-19 With Undetectable Infected Persons,“ *IEEE Transactions on Network Science and Engineering*, Vol. 7, Nr. 4, S. 3279–3294, 2020.
- [8] J. R. C. Piqueira und V. O. Araujo, „A modified epidemiological model for computer viruses,“ *Applied Mathematics and Computation*, Vol. 213, Nr. 2, S. 355–360, 2009. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0096300309002379>
- [9] H. S. Rodrigues, „Application of SIR epidemiological model: new trends,“ 2016. [Online]. Verfügbar: <https://doi.org/10.48550/arXiv.1611.02565>
- [10] H. W. Hethcote, „The Mathematics of Infectious Diseases,“ *SIAM Review*, Vol. 42, Nr. 4, S. 599–653, 2000. [Online]. Verfügbar: <http://www.jstor.org/stable/2653135>
- [11] R. Schlickeiser und M. Kröger, „Analytical Modeling of the Temporal Evolution of Epidemics Outbreaks Accounting for Vaccinations,“ *Physics*, Vol. 3, Nr. 2, S. 386–426, 2021. [Online]. Verfügbar: <https://www.mdpi.com/2624-8174/3/2/28>

- [12] A. Kaddar, A. Abta, und A. H. Talibi, „A comparison of delayed SIR and SEIR epidemic models,” *Nonlinear Analysis: Modelling and Control*, Vol. 16, Nr. 2, S. 181–190, Apr. 2011. [Online]. Verfügbar: <https://www.journals.vu.lt/nonlinear-analysis/article/view/14104>
- [13] M. Niazi und A. Hussain, „Agent-based computing from multi-agent systems to agent-based Models: a visual survey,” *Scientometrics*, Vol. 89, S. 479–499, 11 2011.
- [14] Y. Kubera, P. Mathieu, und S. Picault, „Everything can be Agent!” in *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Serie Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), W. van der Hoek, G. Kaminka, Y. Lespérance, M. Luck, und S. Sen, Hrsgg. Toronto, Canada: International Foundation for Autonomous Agents and Multiagent Systems, Mai 2010, S. 1547–1548. [Online]. Verfügbar: <https://hal.science/hal-00584364>
- [15] J. Hu, A. E. Turgut, B. Lennox, und F. Arvin, „Robust Formation Coordination of Robot Swarms With Nonlinear Dynamics and Unknown Disturbances: Design and Experiments,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 69, Nr. 1, S. 114–118, 2022.
- [16] A. Vatankhah Barenji und R. Vatankhah Barenji, „Improving multi-agent manufacturing control system by indirect communication based on ant agents,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 231, Nr. 6, S. 447–458, 2017.
- [17] J. F. Traniello, „Foraging strategies of ants,” *Annual review of entomology*, Vol. 34, Nr. 1, S. 191–210, 1989.
- [18] A. Dorri, S. S. Kanhere, und R. Jurdak, „Multi-Agent Systems: A Survey,” *IEEE Access*, Vol. 6, S. 28 573–28 593, 2018.
- [19] P. Leito und S. Karnouskos, *Industrial Agents: Emerging Applications of Software Agents in Industry*, 1. Aufl. NLD: Elsevier Science Publishers B. V., 2015.
- [20] P. Révész, *Random walk in random and non-random environments*. World Scientific, 2013.
- [21] P. Bovet und S. Benhamou, „Spatial analysis of animals’ movements using a correlated random walk model,” *Journal of Theoretical Biology*, Vol. 131, Nr. 4, S. 419–433, 1988. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0022519388800389>
- [22] R. E. Jones, „Movement Patterns and Egg Distribution in Cabbage Butterflies,” *Journal of Animal Ecology*, Vol. 46, Nr. 1, S. 195–212, 1977, im Anhang. [Online]. Verfügbar: <http://www.jstor.org/stable/3956>
- [23] E. Renshaw und R. Henderson, „The correlated random walk,” *Journal of Applied Probability*, Vol. 18, Nr. 2, S. 403–414, 1981.
- [24] P. M. Kareiva und N. Shigesada, „Analyzing insect movement as a correlated random walk,” *Oecologia*, Vol. 56, Nr. 2, S. 234–238, Feb 1983. [Online]. Verfügbar: <https://doi.org/10.1007/BF00379695>



- [25] P. Karlson und M. Lüscher, „‘Pheromones’: a New Term for a Class of Biologically Active Substances,” *Nature*, Vol. 183, Nr. 4653, S. 55–56, Jan 1959. [Online]. Verfügbar: <https://doi.org/10.1038/183055a0>
- [26] J. H. Law und F. E. Regnier, „Pheromones,” *Annual review of biochemistry*, Vol. 40, Nr. 1, S. 533–548, 1971.
- [27] M. S. Blum und E. O. Wilson, „The anatomical source of trail substances in formicine ants,” *Psyche*, Vol. 71, Nr. 1, S. 28–31, 1964.
- [28] J. S. Elkinton und R. T. Cardé, „Odor Dispersion,” in *Chemical Ecology of Insects*, W. J. Bell und R. T. Cardé, Hrsgg. Boston, MA: Springer US, 1984, S. 73–91. [Online]. Verfügbar: [https://doi.org/10.1007/978-1-4899-3368-3\\_3](https://doi.org/10.1007/978-1-4899-3368-3_3)
- [29] W. H. Bossert und E. O. Wilson, „The analysis of olfactory communication among animals,” *Journal of Theoretical Biology*, Vol. 5, Nr. 3, S. 443–469, 1963. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/0022519363900894>
- [30] Y. Fares, P. J. Sharpe, und C. E. Magnuson, „Pheromone dispersion in forests,” *Journal of Theoretical Biology*, Vol. 84, Nr. 2, S. 335–359, 1980. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0022519380800105>
- [31] A. W. FERGUSON und J. B. FREE, „Queen pheromone transfer within honeybee colonies,” *Physiological Entomology*, Vol. 5, Nr. 4, S. 359–366, 1980. [Online]. Verfügbar: <https://resjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-3032.1980.tb00245.x>
- [32] W. S. Leal, „Pheromone reception,” *The Chemistry of Pheromones and Other Semiochemicals II: -/-*, S. 1–36, 2005.
- [33] J. Kohl, P. Huoviala, und G. S. Jefferis, „Pheromone processing in *Drosophila*,” *Current Opinion in Neurobiology*, Vol. 34, S. 149–157, 2015, Molecular biology of sensation. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0959438815001038>
- [34] E. David Morgan, „Trail pheromones of ants,” *Physiological Entomology*, Vol. 34, Nr. 1, S. 1–17, 2009. [Online]. Verfügbar: <https://resjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-3032.2008.00658.x>
- [35] F. J. Verheggen, E. Haubruge, und M. C. Mescher, „Chapter Nine - Alarm Pheromones—Chemical Signaling in Response to Danger,” in *Pheromones*, Serie Vitamins & Hormones, G. Litwack, Hrsg. Academic Press, 2010, Vol. 83, S. 215–239. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0083672910830092>
- [36] R. Benton, „Sensitivity and specificity in *Drosophila* pheromone perception,” *Trends in Neurosciences*, Vol. 30, Nr. 10, S. 512–519, 2007. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0166223607001853>
- [37] R. Fujisawa, S. Dobata, K. Sugawara, und F. Matsuno, „Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance,” *Swarm Intelligence*, Vol. 8, Nr. 3, S. 227–246, Sep 2014. [Online]. Verfügbar: <https://doi.org/10.1007/s11721-014-0097-z>



- [38] R. Fujisawa, H. Imamura, T. Hashimoto, und F. Matsuno, „Communication using pheromone field for multiple robots,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, S. 1391–1396.
- [39] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, und S. Honiden, „Performance of Pheromone Model for Predicting Traffic Congestion,” in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Serie AAMAS '06. New York, NY, USA: Association for Computing Machinery, 2006, S. 73–80. [Online]. Verfügbar: <https://doi.org/10.1145/1160633.1160642>
- [40] W. L. Leong, J. Cao, S. Huang, und R. Teo, „Pheromone-Based Approach for Scalable Task Allocation,” in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, S. 220–227.
- [41] J. F. A. Traniello, „Foraging Strategies of Ants,” *Annual Review of Entomology*, Vol. 34, Nr. 1, S. 191–210, 1989. [Online]. Verfügbar: <https://doi.org/10.1146/annurev.en.34.010189.001203>
- [42] L. Panait und S. Luke, „Ant foraging revisited,” in *proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, 2004.
- [43] M. Dorigo, M. Birattari, und T. Stutzle, „Ant colony optimization,” *IEEE computational intelligence magazine*, Vol. 1, Nr. 4, S. 28–39, 2006.
- [44] S. Dübener und J. Wagner, „Quick introduction to complex programming in schools with the educational robot Dezibot - Evaluation and Review,” in *2nd International Conference on Information Technology, Engineering, Robotics, Innovation Research and Structural Design*, 08 2021.
- [45] S. U. Dübener, A. A. Morgner, H. F. Haupt, M. H. Volk, C. C. Fischer, S. K. Langner, und A. Jacker, „Gegenüberstellung von kostengünstigen Robotern als Lernobjekte für Schulen,” in *INFORMATIK 2017*, M. Eibl und M. Gaedke, Hrsgg. Gesellschaft für Informatik, Bonn, 2017, S. 2461–2472.
- [46] S. Dübener, A. Jacker, A. Morgner, H. Haupt, und J. Wagner, „Dezibot2 - The Wi-Fi based, inexpensive, and small educational mobile robot,” in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, 08 2018.
- [47] G. O. LÓPEZ-RIQUELME, E. A. Malo, L. CRUZ-LÓPEZ, und M. L. FANJUL-MOLES, „Antennal olfactory sensitivity in response to task-related odours of three castes of the ant *Atta mexicana* (hymenoptera: formicidae),” *Physiological Entomology*, Vol. 31, Nr. 4, S. 353–360, 2006.
- [48] N. Collier und M. North, „Parallel agent-based simulation with Repast for High Performance Computing,” *SIMULATION*, Vol. 89, Nr. 10, S. 1215–1235, 2013. [Online]. Verfügbar: <https://doi.org/10.1177/0037549712462620>
- [49] M. J. North, N. T. Collier, J. Ozik, E. R. Tataro, C. M. Macal, M. Bragen, und P. Sydelko, „Complex adaptive systems modeling with Repast Symphony,” *Complex Adaptive Systems Modeling*, Vol. 1, Nr. 1, S. 3, Mar 2013. [Online]. Verfügbar: <https://doi.org/10.1186/2194-3206-1-3>

- [50] N. Collier, „Repast: An extensible framework for agent simulation,” *The University of Chicago’s Social Science Research*, Vol. 36, S. 2003, 2003. [Online]. Verfügbar: <http://www2.econ.iastate.edu/tesfatsi/RePastTutorial.Collier.pdf>
- [51] L. Clarke, I. Glendinning, und R. Hempel, „The MPI Message Passing Interface Standard,” in *Programming Environments for Massively Parallel Distributed Systems*, K. M. Decker und R. M. Rehmann, Hrsgg. Basel: Birkhäuser Basel, 1994, S. 213–218.
- [52] M. Allen, D. Poggiali, K. Whitaker, T. Marshall, J. van Langen, und R. Kievit, „Raincloud plots: a multi-platform tool for robust data visualization [version 2; peer review: 2 approved],” *Wellcome Open Research*, Vol. 4, Nr. 63, 2021. [Online]. Verfügbar: <https://wellcomeopenresearch.org/articles/4-63>
- [53] V. Parma, K. Ohla, M. G. Veldhuizen, M. Y. Niv, C. E. Kelly, A. J. Bakke, K. W. Cooper, C. Bouysset, N. Pirastu, M. Dibattista, R. Kaur, M. T. Liuzza, M. Y. Pepino, V. Schöpf, V. Pereda-Loth, S. B. Olsson, R. C. Gerkin, P. Rohlfs Domínguez, J. Albayay, M. C. Farruggia, S. Bhutani, A. W. Fjaeldstad, R. Kumar, A. Menini, M. Bensafi, M. Sandell, I. Konstantinidis, A. Di Pizio, F. Genovese, L. Öztürk, T. Thomas-Danguin, J. Frasnelli, S. Boesveldt, Ö. Saatci, L. R. Saraiva, C. Lin, J. Golebiowski, L.-D. Hwang, M. H. Ozdener, M. D. Guàrdia, C. Laudamiel, M. Ritchie, J. Havlíček, D. Pierron, E. Roura, M. Navarro, A. A. Nolden, J. Lim, K. L. Whitcroft, L. R. Colquitt, C. Ferdenzi, E. V. Brindha, A. Altundag, A. Macchi, A. Nunez-Parra, Z. M. Patel, S. Fiorucci, C. M. Philpott, B. C. Smith, J. N. Lundström, C. Mucignat, J. K. Parker, M. van den Brink, M. Schmuker, F. P. S. Fischmeister, T. Heinbockel, V. D. C. Shields, F. Faraji, E. Santamaría, W. E. A. Fredborg, G. Morini, J. K. Olofsson, M. Jalessi, N. Karni, A. D’Errico, R. Alizadeh, R. Pellegrino, P. Meyer, C. Huart, B. Chen, G. M. Soler, M. K. Alwashahi, A. Welge-Lüssen, J. Freiherr, J. H. B. de Groot, H. Klein, M. Okamoto, P. B. Singh, J. W. Hsieh, G. G. Author, D. R. Reed, T. Hummel, S. D. Munger, und J. E. Hayes, „More Than Smell—COVID-19 Is Associated With Severe Impairment of Smell, Taste, and Chemesthesis,” *Chemical Senses*, Vol. 45, Nr. 7, S. 609–622, 06 2020. [Online]. Verfügbar: <https://doi.org/10.1093/chemse/bjaa041>
- [54] E. Meyerowitz, A. Richterman, R. Gandhi, und P. Sax, „Transmission of SARS-CoV-2: A Review of Viral, Host, and Environmental Factors,” *Annals of Internal Medicine*, Vol. 174, Nr. 1, S. 69–79, 2021, PMID: 32941052. [Online]. Verfügbar: <https://doi.org/10.7326/M20-5008>
- [55] R. Schlickeiser und M. Kröger, „Analytical Modeling of the Temporal Evolution of Epidemics Outbreaks Accounting for Vaccinations,” *Physics*, Vol. 3, Nr. 2, S. 386–426, 2021. [Online]. Verfügbar: <https://www.mdpi.com/2624-8174/3/2/28>

## Abbildungsverzeichnis

2.1	Diagramm zu den Gruppen des <i>SEIR</i> -Modells . . . . .	8
2.2	Verlaufs-Graphen eines beispielhaften <i>SEIR</i> -Modells . . . . .	9
2.3	Gegenüberstellung von <i>Random Walk</i> und <i>Correlated Random Walk</i> . . . . .	11
2.4	Seitenansicht von einem <i>DeziBot<sub>3</sub></i> . . . . .	15
3.1	Aufbau des Spielfeldes . . . . .	17
3.2	Basis-Bewegungs-Muster der Agenten . . . . .	21
3.3	Verhalten der Agenten an den Spielfeld-Rändern . . . . .	22
3.4	Darstellungen der Lese- und Schreibfelder eines Agenten für die Pheromon- Interaktion . . . . .	24
3.5	Allgemeines Diagramm zum Aufbau der Ressourcen-Suche von Agenten . . . . .	25
3.6	Diagramm zu den <i>SEIR</i> -Zuständen der Agenten . . . . .	26
3.7	Infizierbarkeit der Agenten . . . . .	27
3.8	Übersicht des Spielfeldes mit Agenten in der Simulation als <i>UML</i> -Diagramm . . . . .	29
4.1	Beispiel zum Verlauf des Infektionsgeschehens in der Simulation . . . . .	32
4.2	Verteilung der Werte von $I_{max}$ für alle Simulationsdurchläufe der vier Haupt- experimente. . . . .	36
4.3	Verteilung der Werte von $R_{max}$ für alle Simulationsdurchläufe der vier Haupt- experimente . . . . .	37
4.4	Verteilung der Werte von $I_{max}$ für alle Simulationsdurchläufe der drei Zusatz- experimente. . . . .	39
4.5	Verteilung der Werte von $R_{max}$ für alle Simulationsdurchläufe der drei Zu- satzexperimente . . . . .	40

## Tabellenverzeichnis

0.1	Genutzte mathematische Symbole . . . . .	3
0.2	Darstellungen von Diagramm-Elementen . . . . .	4
0.3	Namensgebung der Simulationsparameter . . . . .	4
4.1	Die Ergebnisse der Ausgangsexperimente( <i>ID</i> 0 und 1) in absoluten Zahlen- werten . . . . .	34
4.2	Ergebnisse der Hauptexperimente. . . . .	35
4.3	Ergebnisse der Zusatzexperimente. . . . .	38

## Algorithmen-Verzeichnis

3.1	<i>Update</i> -Funktion für das Ressourcen-Modell . . . . .	17
3.2	Funktionen zum setzen der Pheromon-Intensität . . . . .	19
3.3	Algorithmen für das Schreiben von Pheromon-Intensitäten . . . . .	23
3.4	Algorithmus für die Ermittlung der Position, des ersten Pheromons im Sichtfeld des Agenten $a$ . . . . .	23
3.5	Prozess zur Erstellung und Initialisierung aller Agenten zu Beginn der Simulation. . . . .	28

## Anlagenverzeichnis

1	Inhaltsübersicht der beigefügte Dateien. . . . .	53
2	Vollständige Tabelle aller Parameter für das Simulationsmodell allgemein. . .	54
3	Vollständige Tabelle aller Parameter für das Spielfeldmodell. . . . .	54
4	Vollständige Tabelle aller Parameter für das Pheromon-Modell. . . . .	54
5	Vollständige Tabelle aller Parameter für das Ressourcenmodell. . . . .	54
6	Vollständige Tabelle aller Parameter für das Bewegungsmodell der Agenten.	55
7	Vollständige Tabelle aller Parameter für das Ressourcenmodell der Agenten.	55
8	Vollständige Tabelle aller Parameter für das Infektionsmodell der Agenten. .	56
9	Vollständige Tabelle aller Parameter für die Pheromon-Interaktion der Agenten.	57
10	Vollständige Tabelle aller Parameter für die initialen <i>SEIR</i> -Zustände der Agenten. . . . .	58
11	Genutzte Hilfsfunktionen für die folgenden Algorithmen der Ressourcen-Suche.	59
12	Ressource-Suche-Algorithmus im Zustand <i>Explore</i> . . . . .	59
13	Ressource-Suche-Algorithmus im Zustand <i>Follow</i> . . . . .	60
14	Ressource-Suche-Algorithmus im Zustand <i>Draw</i> . . . . .	60
15	Ressource-Suche-Algorithmus im Zustand <i>Wait</i> . . . . .	60
16	Gemittelter Simulationsverlauf: Ausgangswerte ohne Krankheitsausbreitung ( <i>ID</i> : 0). . . . .	61
17	Gemittelter Simulationsverlauf: Ausgangswerte mit Krankheitsausbreitung ( <i>ID</i> : 1). . . . .	62
18	Gemittelter Simulationsverlauf: Verhaltensanpassung, dass symptomatische Agenten stehen bleiben ( <i>ID</i> : 2). . . . .	63
19	Gemittelter Simulationsverlauf: Verhaltensanpassung, dass infizierbare Agenten den infektiösen Agenten ausweichen ( <i>ID</i> : 3). . . . .	64
20	Gemittelter Simulationsverlauf: Kombination der zwei Verhaltensanpassungen ( <i>ID</i> : 4). . . . .	65
21	Gemittelter Simulationsverlauf: Experiment zur <i>Gruppierung</i> der infektiösen Agenten mithilfe vom Warn-Pheromon ( <i>ID</i> : 5). . . . .	66
22	Gemittelter Simulationsverlauf: Infektiöse können „nicht riechen“ (Reagieren nicht auf Indikator-Pheromon für Ressourcen) ( <i>ID</i> : 6). . . . .	67
23	Gemittelter Simulationsverlauf: Infektiöse „riechen falsch“ (Reagieren nicht auf Indikatoren und folgen zufällig Phantom-Pheromonen) ( <i>ID</i> : 7). . . . .	68

## Anlagen

Zu der Bachelorarbeit gehört eine Sammlung von Dateien. Diese ist auf den *Servern* der Uni-Leipzig unter: <https://git.informatik.uni-leipzig.de/pr83zyfu/bachelor-files> zu finden. Die Sammlung besteht aus folgenden Inhalten:

Ordner	Inhalt
<i>/thesis</i>	Diese Bachelorarbeit als <i>PDF</i>
<i>/simulation</i>	Quellcode der Simulations-Implementierung
<i>/visual</i>	Quellcode von genutzten <i>Python</i> -Skripten zur Erstellung von Graphiken und Videos für die Arbeit
<i>/experiments</i>	Rohdaten zu den einzelnen Experimenten. Die Daten zum Experiment mit der <i>ID i</i> sind jeweils im Unterordner <i>/output_i</i> zu finden. <i>/output_-1</i> enthält die Daten zum Experiment zur Effektivität der Ressourcen-Suche (siehe <a href="#">Abschnitt 4.6</a> )
<i>/videos</i>	Beispiel-Videos zu den Modellen (siehe <a href="#">Abschnitt 3</a> ) und jeweils die Rohdaten der Simulationen

Der Ordner */videos* enthält folgende Video-Dateien:

Datei	Beschreibung
<i>/move.mp4</i>	Darstellung zum Basis-Bewegungsschema der Agenten. Zwei Agenten bewegen sich im <i>Correlated Random Walk</i> und prallen von Wänden ab.
<i>/resources.mp4</i>	Beispiel zur Ressourcen-Suche. Die Agenten sind entsprechend ihrem Zustand im Such-Prozess dargestellt: Punkt ( <i>Explore</i> ), Quadrat ( <i>Wait</i> ), Dreieck ( <i>Draw</i> ), Kreis ( <i>Follow</i> ). Das Indikator-Pheromon und die Ressourcen-Quellen sind in grün eingezeichnet.
<i>/seir.mp4</i>	Beispielhaftes Infektionsgeschehen mit 100 Agenten, ohne Ressourcen-Suche.

**Anlage 1:** Inhaltsübersicht der beigefügten Dateien.



Parameter	Grundmenge	Standard	Beschreibung
$x_{random-seed}$	$\mathbb{N}_0$	0	<i>Seed</i> für die Zufallsfunktionen der Simulation
$t_{max}$	$\mathbb{N}_0$	2000	Obere Zeitgrenze für die Dauer der Simulation (maximale Simulationsschritte)

**Anlage 2:** Vollständige Tabelle aller Parameter für das Simulationsmodell allgemein.

Parameter	Grundmenge	Standard	Beschreibung
$w_{arena}$	$\mathbb{E} \cap \mathbb{N}$	100	Breite des Spielfeldes
$h_{arena}$	$\mathbb{E} \cap \mathbb{N}$	100	Höhe des Spielfeldes

**Anlage 3:** Vollständige Tabelle aller Parameter für das Spielfeldmodell.

Parameter	Grundmenge	Standard	Beschreibung
$t_{resource-indicator-lifetime}$	$\mathbb{N}_0$	250	Lebenszeit des Ressourcen-Indikator-Pheromons
$t_{warning-lifetime}$	$\mathbb{N}_0$	100	Lebenszeit des Warnungs-Pheromon

**Anlage 4:** Vollständige Tabelle aller Parameter für das Pheromon-Modell.

Parameter	Grundmenge	Standard	Beschreibung
$c_{resources}$	$\mathbb{N}_0$	3	Anzahl an Ressourcen-Quellen, die zu jeder gegebenen Zeit auf dem Spielfeld existieren.
$s_{resources}$	$\mathbb{N}_0$	10	Größe der Ressourcen-Quellen

**Anlage 5:** Vollständige Tabelle aller Parameter für das Ressourcenmodell.

Parameter	Grundmenge	Standard	Beschreibung
$v_{agent}$	$\mathbb{R}^+$	1	(Maximale) Bewegungsgeschwindigkeit der Agenten
$p_{crw}$	$[0, 1]$	1/2	Wahrscheinlichkeit, dass ein Agent im <i>CRW</i> ohne Drehung weiter geradeaus läuft
$\theta_{crw-\delta}$	$\mathbb{R}^+$	1/2	Winkel in Radianten, um den ein Agent im <i>CRW</i> zu einem Zeitschritt sich maximal dreht

**Anlage 6:** Vollständige Tabelle aller Parameter für das Bewegungsmodell der Agenten.

Parameter	Grundmenge	Standard	Beschreibung
$r_{resource-detect}$	$\mathbb{R}^+$	2	Radius, in dem ein Agent Ressourcen-Quellen wahrnehmen kann
$r_{resource-collect}$	$\mathbb{R}^+$	2	Radius, in dem ein Agent Ressourcen aus Ressourcen-Quellen einsammeln kann.
$t_{resource-collect}$	$\mathbb{N}_0$	100	Zeit, die ein Agent für das Einsammeln einer Ressource benötigt.

**Anlage 7:** Vollständige Tabelle aller Parameter für das Ressourcenmodell der Agenten.

Parameter	Grundmenge	Standard	Beschreibung
$p_{infect}$	$[0, 1]$	1/10	Wahrscheinlichkeit, dass ein uninfizierter Agent, in Reichweite von einem infizierenden Agent infiziert wird
$r_{infect}$	$\mathbb{R}^+$	5	Radius, in dem infizierende Agenten andere Agenten infizieren können
$t_{exposed}$	$\mathbb{N}_0$	200	Zeit, die ein Agent im <i>SEIR</i> -Zustand <i>Exposed</i> bleibt
$t_{infectious}$	$\mathbb{N}_0$	200	Zeit, die ein Agent im <i>SEIR</i> -Zustand <i>Infectious</i> bleibt
$t_{realization}$	$\mathbb{N}_0$	0	Zeit, die ein Agent benötigt bis er nach dem Übergang in den <i>SEIR</i> -Zustand <i>Infectious</i> realisiert, dass er infizierend ist
$v_{agent-symptom}$	$\mathbb{R}^+$	1/2	Geschwindigkeit eines symptomatischen Agenten

**Anlage 8:** Vollständige Tabelle aller Parameter für das Infektionsmodell der Agenten.

Parameter	Grundmenge	Standard	Beschreibung
$l_{resource-indicator}$	$\mathbb{R}^+$	40	Die Länge der Linie, die eine Agent beim Fund einer Ressourcen-Quelle als Indikator malt.
$h_{pheromone-detect}$	$\mathbb{N}_0$	1	Die Höhe des Sichtfeldes, in dem ein Agent Pheromone vor ihm erkennen kann.
$w_{pheromone-detect}$	$\mathbb{N}_0 \cap \mathbb{O}$	3	Die Breite des Sichtfeldes, in dem ein Agent Pheromone vor ihm erkennen kann.
$h_{pheromone-draw}$	$\mathbb{N}_0$	2	Die Höhe des Schreibfeldes, in dem ein Agent Pheromone hinter ihm schreiben kann.
$w_{pheromone-draw}$	$\mathbb{N}_0 \cap \mathbb{O}$	4	Die Breite des Schreibfeldes, in dem ein Agent Pheromone hinter ihm schreiben kann.
$p_{phantom-start}$	$[0, 1]$	1/10	Die Wahrscheinlichkeit, dass ein symptomatischer Agent in Experiment 7 beginnt ein Phantom-Pheromon wahrzunehmen, bzw. geradeaus zu laufen.
$p_{phantom-stop}$	$[0, 1]$	2/100	Die Wahrscheinlichkeit, dass ein symptomatischer Agent in Experiment 7 aufhört ein Phantom-Pheromon wahrzunehmen, bzw. geradeaus zu laufen.

**Anlage 9:** Vollständige Tabelle aller Parameter für die Pheromon-Interaktion der Agenten.

Parameter	Grundmenge	Standard	Beschreibung
$c_{initial-susceptible}$	$\mathbb{N}_0$	99	Anzahl der Agenten, die zum Start der Simulation im Zustand <i>Susceptible</i> sind.
$c_{initial-exposed}$	$\mathbb{N}_0$	1	Anzahl der Agenten, die zum Start der Simulation im Zustand <i>Exposed</i> sind.
$c_{initial-infectious}$	$\mathbb{N}_0$	0	Anzahl der Agenten, die zum Start der Simulation im Zustand <i>Infectious</i> sind.
$c_{initial-recovered}$	$\mathbb{N}_0$	0	Anzahl der Agenten, die zum Start der Simulation im Zustand <i>Recovered</i> sind.

**Anlage 10:** Vollständige Tabelle aller Parameter für die initialen *SEIR*-Zustände der Agenten.

```

1: function MOVEFORWARD()
2:   ...                                ▷ Eine einfache Bewegung geradeaus um  $v_{agent}$ 
3: end function

4: function CRW()
5:   ...                                ▷ CRW-Update aus Abschnitt 3.5.1
6: end function

7: function GETUNCOLLECTEDRESOURCE( $a$ )
8:   for  $r \in R$  do
9:     if  $\text{dist}(a, r) \leq r_{resource-detect} \wedge a$  hat  $r$  noch nicht eingesammelt then
10:      return  $r$ 
11:    end if
12:  end for
13:  return NULL
14: end function

15: function SWITCHTOSTATE( $s_{new}$ )
16:  if  $s_{new} = \text{Draw}$  then
17:     $l_{left-to-draw} \leftarrow l_{resource-indicator}$ 
18:    TURNAROUND()
19:  end if
20:  if  $s_{new} = \text{Wait}$  then
21:    Sammle eine Ressource aus GETUNCOLLECTEDRESOURCE( $a$ )
22:     $b_{found-via-indicator} \leftarrow (s_{current} = \text{Follow})$ 
23:     $t_{wait} \leftarrow t_{resource-collect}$ 
24:  end if
25:
26:   $s_{current} \leftarrow s_{new}$ 
27: end function

```

**Anlage 11:** Genutzte Hilfsfunktionen für die folgenden Algorithmen der Ressourcen-Suche. Für symptomatische Agenten muss in den Bewegungsfunktionen der Wert  $v_{agent}$  durch  $v_{agent-symptom}$  ausgetauscht werden.

```

1: CRW()
2: if GETUNCOLLECTEDRESOURCE( $a$ )  $\neq$  NULL then
3:   SWITCHTOSTATE(Wait)
4: else if FIRSTPHEROMONE( $resInd$ )  $\neq$  NULL then
5:   SWITCHTOSTATE(Follow)
6: end if

```

**Anlage 12:** Vereinfachter Algorithmus für das Agentenverhalten bei der Ressourcen-Suche im *Explore*-Zustand für einen Agenten  $a$ .

```

1: if GETUNCOLLECTEDRESOURCE( $a$ )  $\neq$  NULL then
2:   SWITCHTOSTATE(Wait)
3: end if
4:  $p_{indicator} \leftarrow$  FIRSTPHEROMONE( $resInd$ )
5: if  $p_{indicator} \neq$  NULL then
6:   ROTATETOPOINT( $p_{indicator}$ )
7:   MOVEFORWARD()
8: else
9:   CRW()
10:  SWITCHTOSTATE(Explore)
11: end if

```

**Anlage 13:** Vereinfachter Algorithmus für das Agentenverhalten bei der Ressourcen-Suche im *Follow*-Zustand für einen Agenten  $a$ .

```

1: DRAWPHEROMONE()
2: MOVEFORWARD()
3:  $l_{left-to-draw} \leftarrow l_{left-to-draw} - v_{agent}$ 
4: if  $l_{left-to-draw} \leq 0$  then
5:   SWITCHTOSTATE(Explore)
6: end if

```

**Anlage 14:** Vereinfachter Algorithmus für das Agentenverhalten bei der Ressourcen-Suche im *Draw*-Zustand für einen Agenten  $a$ .

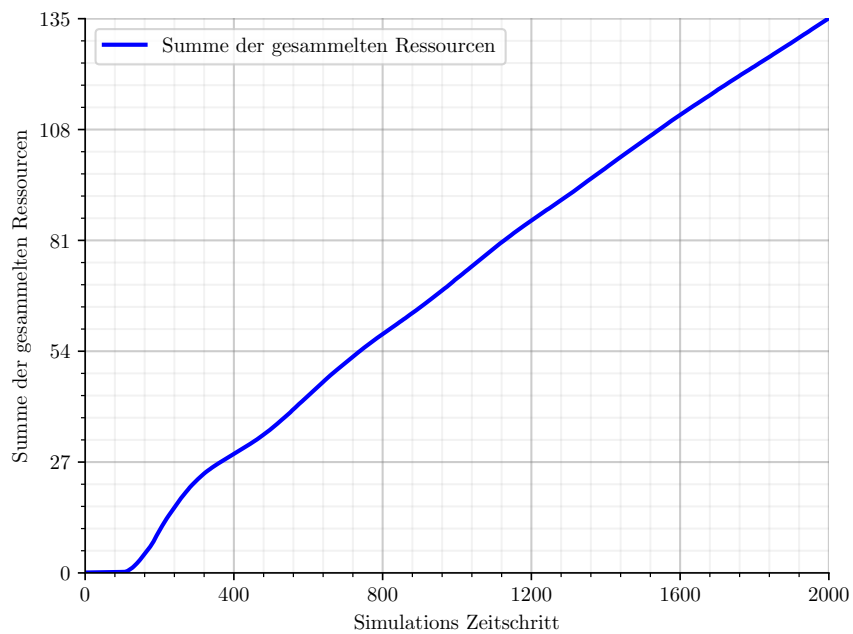
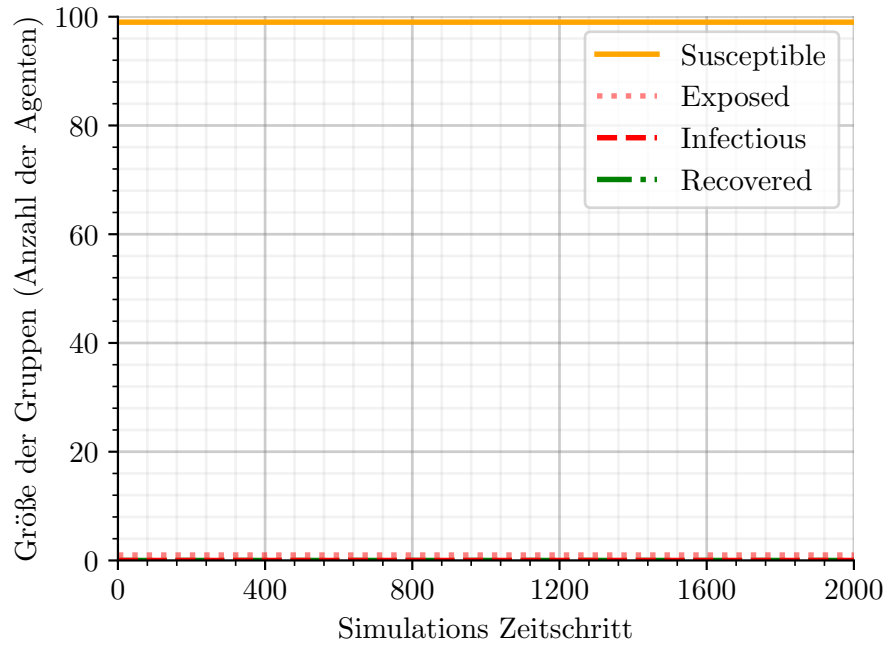
```

1:  $t_{wait} \leftarrow t_{wait} - 1$ 
2: if  $t_{wait} \leq 0$  then
3:   if  $b_{found-via-indicator} =$  FALSE then
4:     SWITCHTOSTATE(Draw)
5:   else
6:     SWITCHTOSTATE(Explore)
7:   end if
8: end if

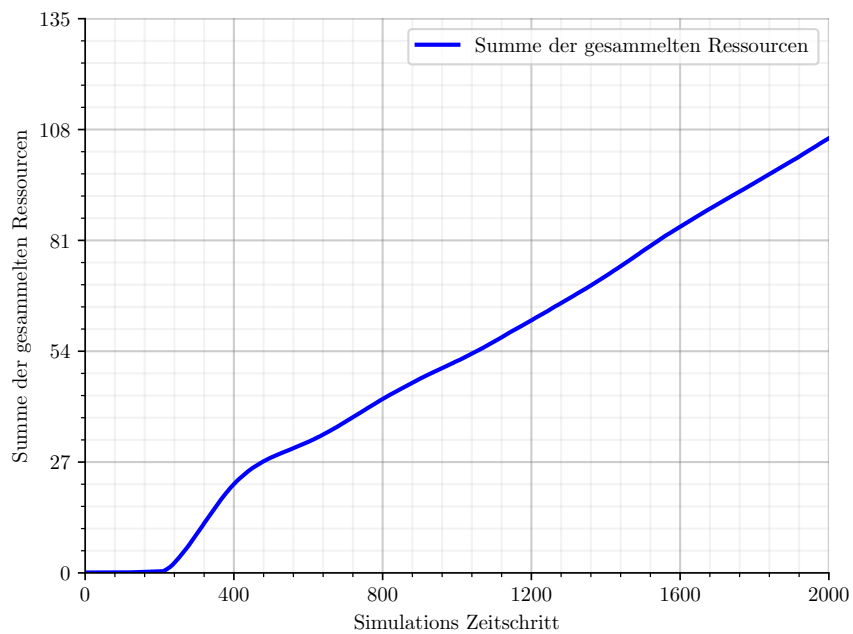
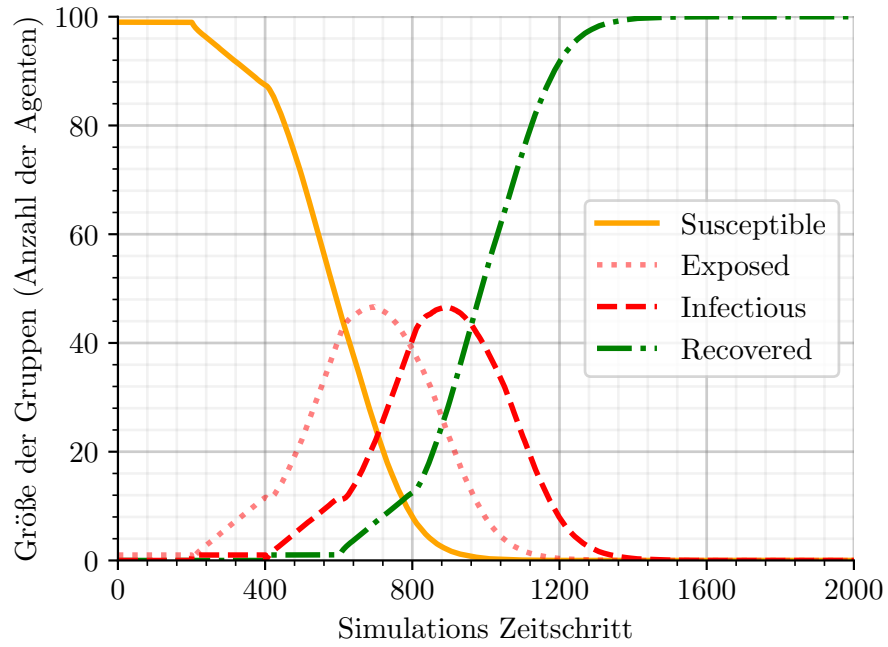
```

**Anlage 15:** Vereinfachter Algorithmus für das Agentenverhalten bei der Ressourcen-Suche im *Wait*-Zustand für einen Agenten  $a$ .

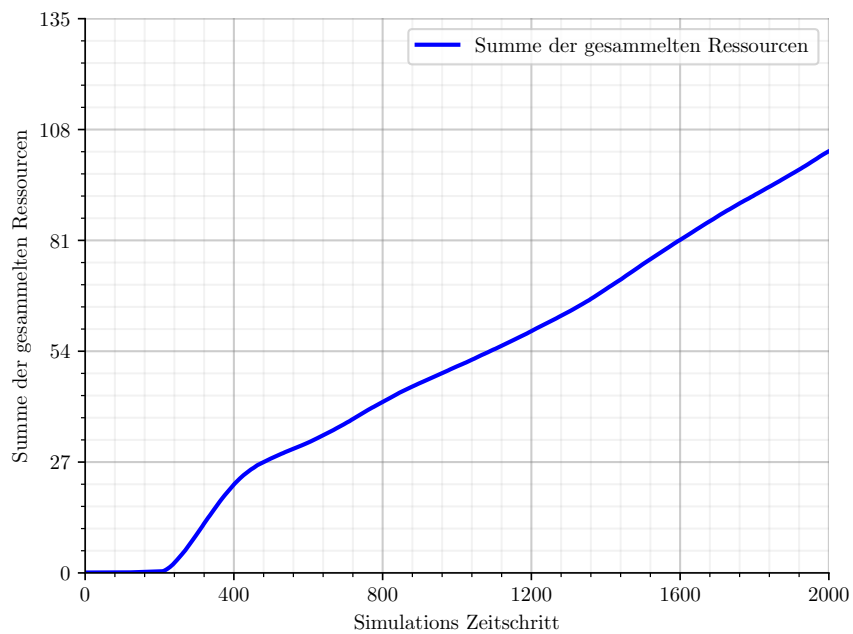
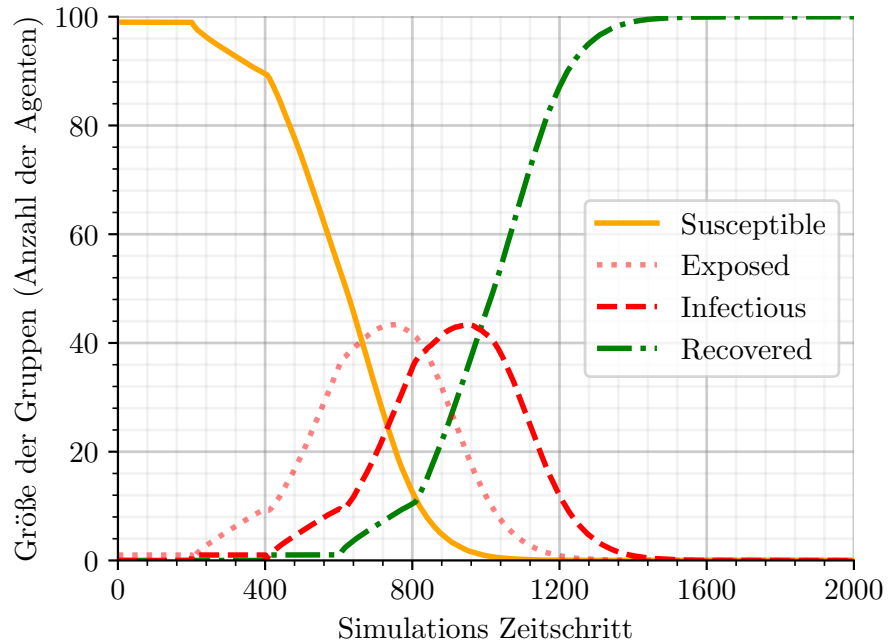




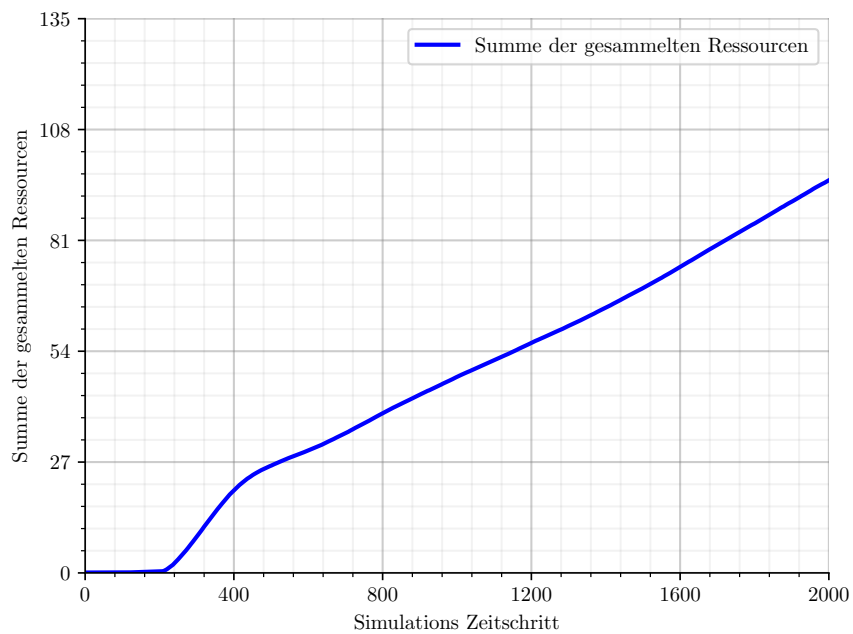
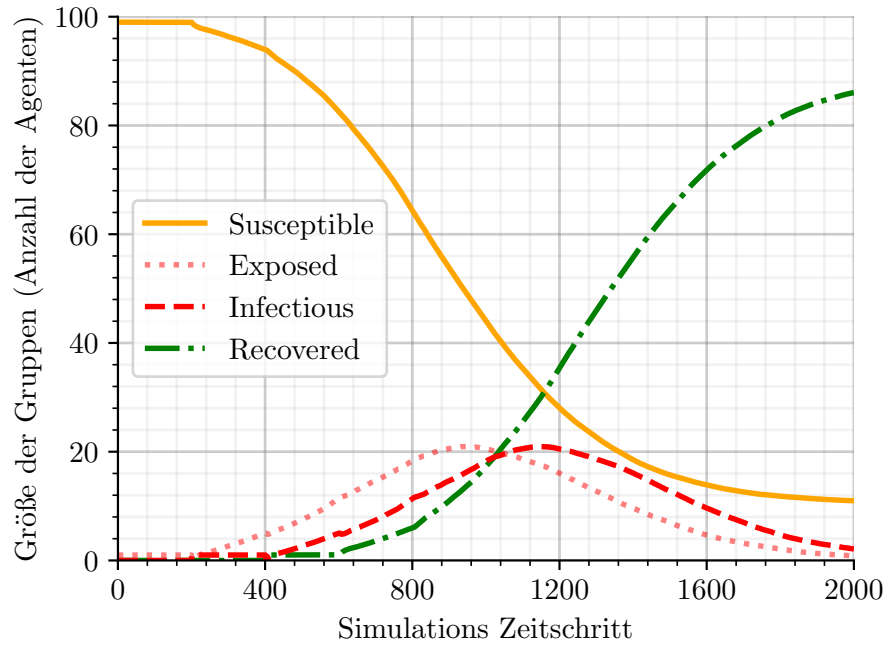
**Anlage 16:** Gemittelter Simulationsverlauf: Ausgangswerte ohne Krankheitsausbreitung (*ID*: 0).



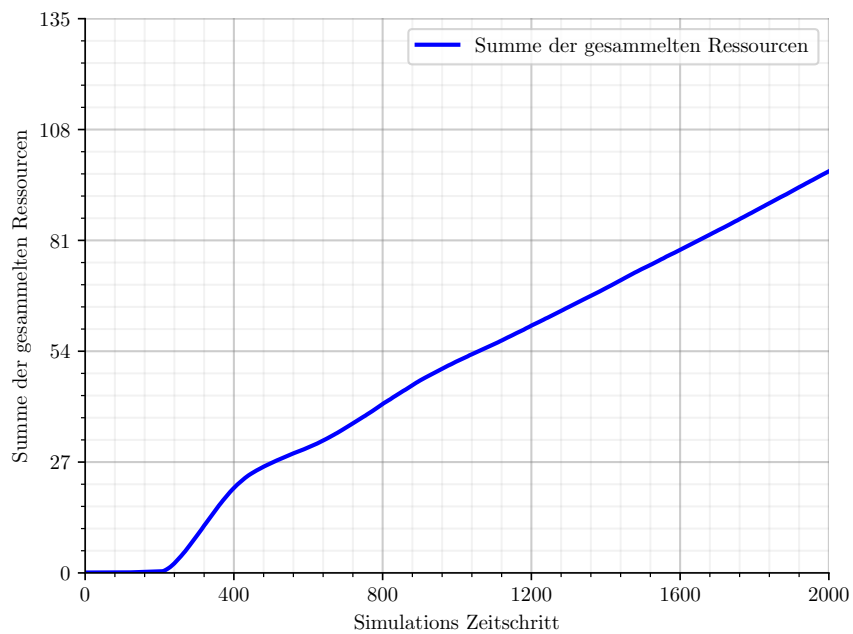
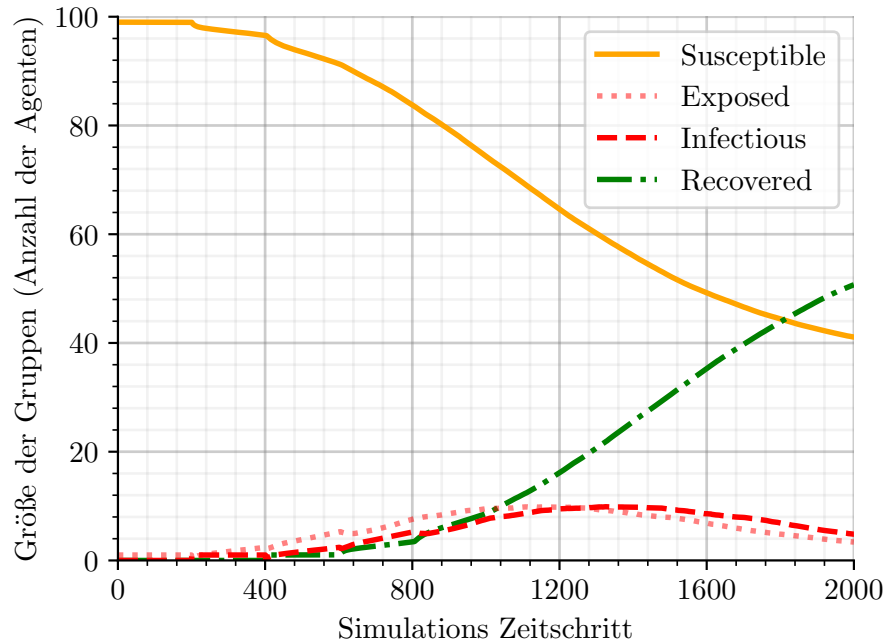
**Anlage 17:** Gemittelter Simulationsverlauf: Ausgangswerte mit Krankheitsausbreitung (*ID*: 1).



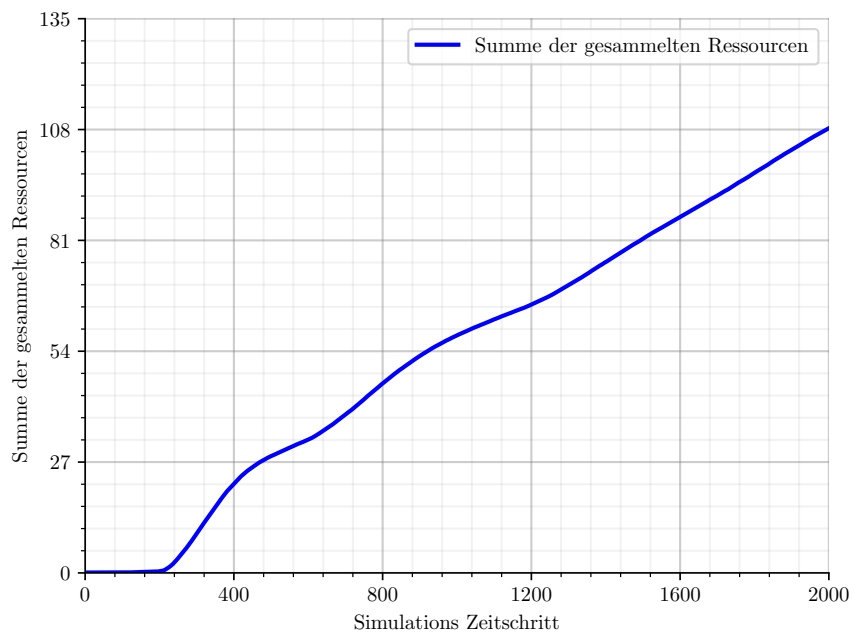
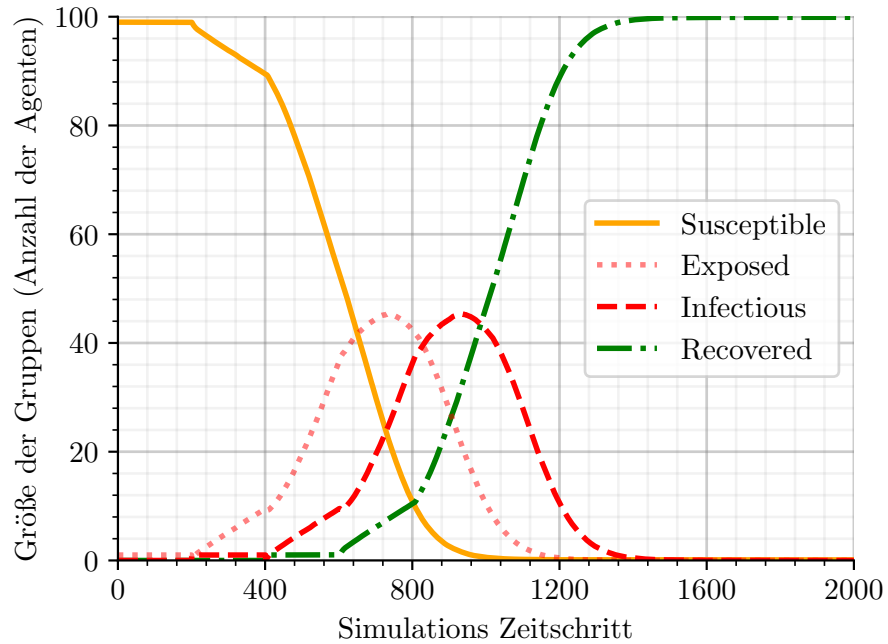
**Anlage 18:** Gemittelter Simulationsverlauf: Verhaltensanpassung, dass symptomatische Agenten stehen bleiben (*ID*: 2).



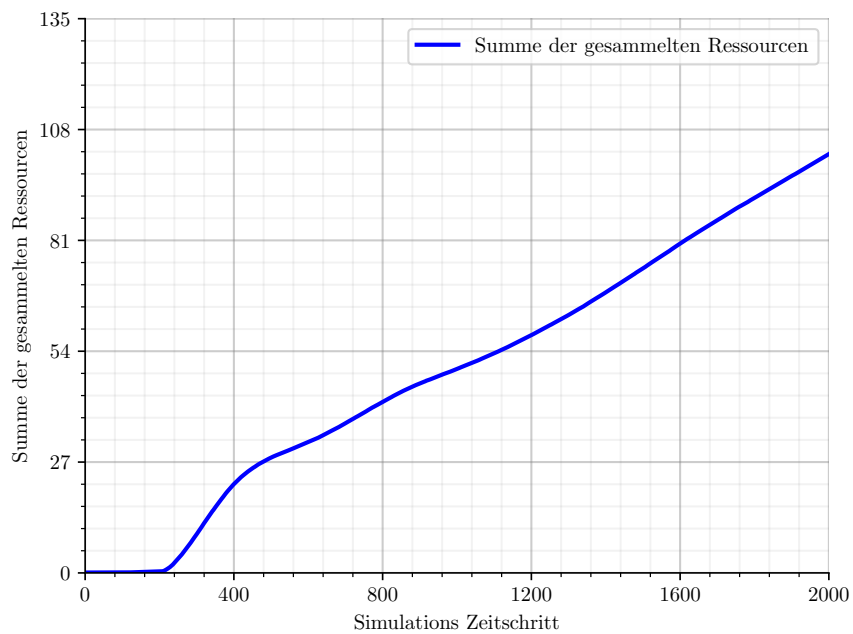
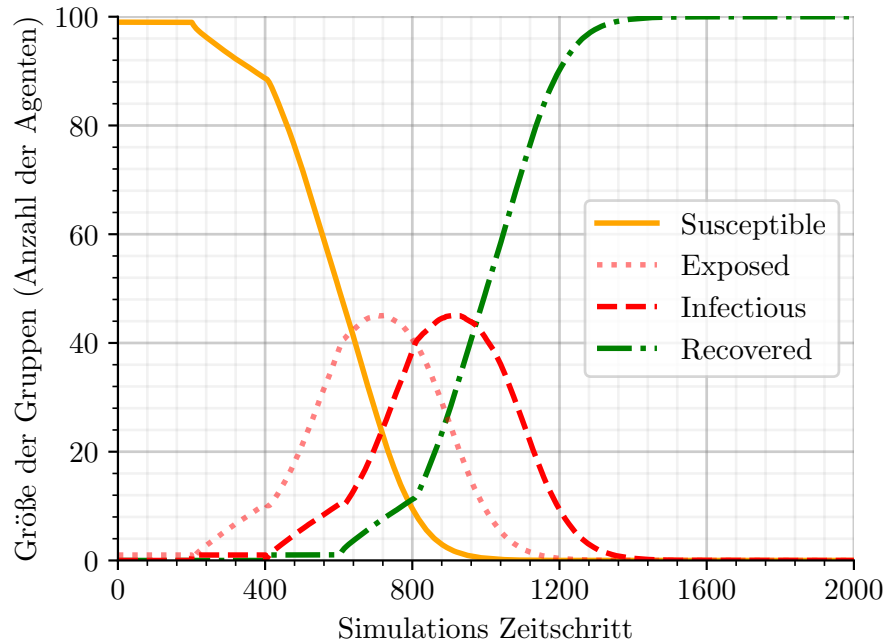
**Anlage 19:** Gemittelter Simulationsverlauf: Verhaltensanpassung, dass infizierbare Agenten den infektiösen Agenten ausweichen (*ID*: 3).



**Anlage 20:** Gemittelter Simulationsverlauf: Kombination der zwei Verhaltensanpassungen (*ID*: 4).

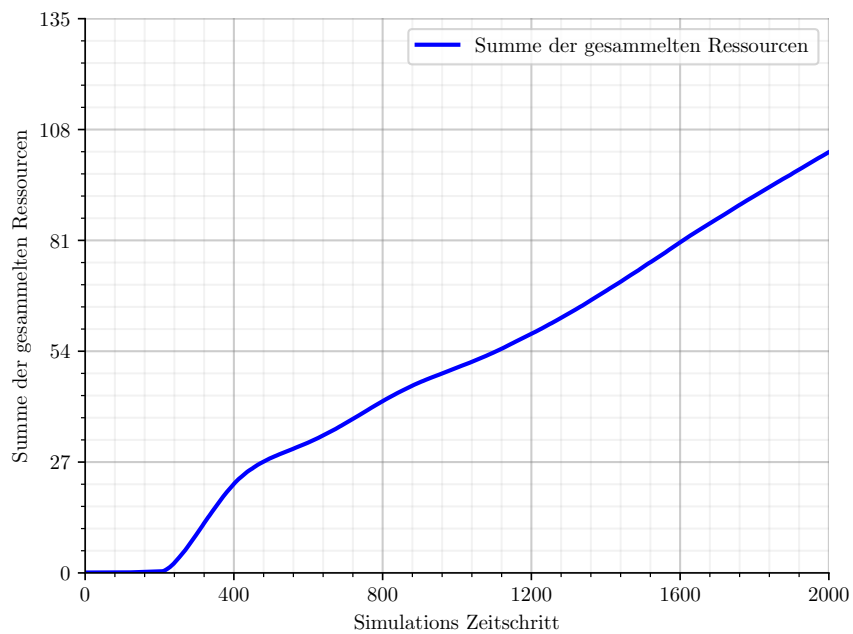
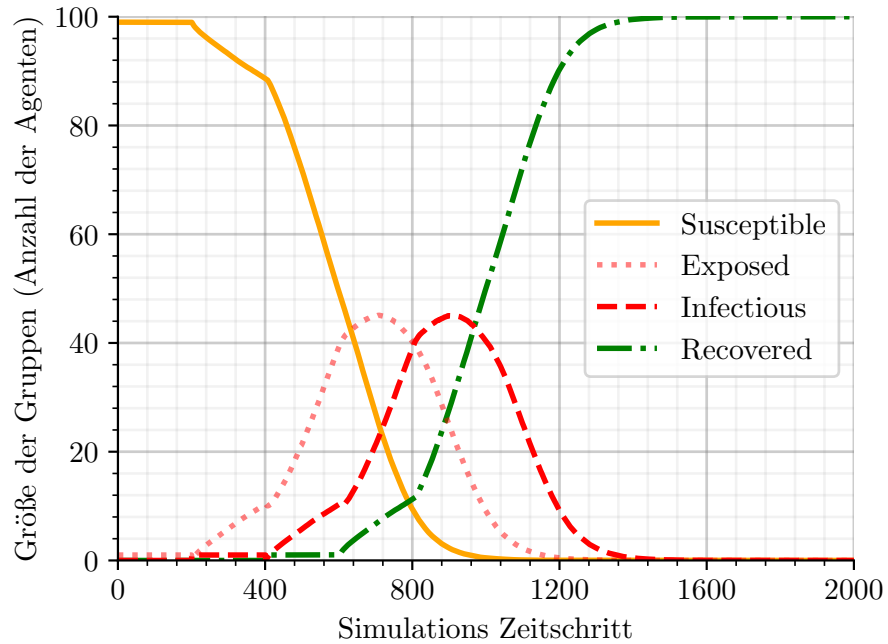


**Anlage 21:** Gemittelter Simulationsverlauf: Experiment zur *Gruppierung* der infektiösen Agenten mithilfe vom Warn-Pheromon (*ID*: 5).



**Anlage 22:** Gemittelter Simulationsverlauf: Infektiöse können „nicht riechen“ (Reagieren nicht auf Indikator-Pheromon für Ressourcen) (ID: 6).





**Anlage 23:** Gemittelter Simulationsverlauf: Infektiöse „riechen falsch“ (Reagieren nicht auf Indikatoren und folgen zufällig Phantom-Pheromonen) (*ID*: 7).

## Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

---

Ort, Datum

---

Unterschrift