

# Optimierung in simulierten biologischen Multiagentensystemen mit Hilfe evolutionärer Verfahren

Alexander Hörnlein, Christoph Oechslein, Frank Puppe

Universität Würzburg  
Lehrstuhl für Informatik VI  
Am Hubland

<http://ki.informatik.uni-wuerzburg.de/>

## ***Einleitung***

Simulationen sind anerkannte Verfahren, um Problemstellungen zu modellieren und zu lösen. An ihnen können unter anderem Planspiele durchgeführt oder mittels Optimierung gute Einstellungen für Modellparameter gefunden werden. Unter Optimierung wird in diesem Papier eine möglichst gute Annäherung eines parametrisierten Systems an den maximalen Wert einer Bewertungsfunktion verstanden.

Auch im Bereich der simulierten biologischen Multiagentensysteme sind diese Verfahren von großem Interesse: Anstatt teure Experimente in der freien Natur, in der die Umgebung erst in kontrollierte Rahmenbedingungen gebracht werden muss, abzuhalten, können im Modell sehr leicht und kostengünstig Änderungen z.B. an der Umwelt simuliert werden. In diesem Zusammenhang ergeben sich auch Optimierungsfragen, wie z.B. welches Verhaltensmodell gemäß einer Bewertung am effizientesten ist.

Aber bevor es möglich ist diese Fragen an ein Modell zu stellen bzw. um von Antworten des Modells Rückschlüsse auf die reale Problemstellung ziehen zu können, muss das Modell valide sein und daher richtig kalibriert werden [Oechs99]. Die Kalibrierung kann dabei auch als ein Optimierungsproblem angesehen werden, dabei werden die zu kalibrierenden Parameter des Agenten gemäß einer Bewertungsfunktion, die das Ein- und Ausgabeverhalten des Systems bzgl. der zu erreichenden Werte vergleicht, optimiert.

In der Literatur wird die Optimierung von simulierten Multiagentensystemen sehr stiefmütterlich behandelt. Optimierung wird meist ‚nur‘ im Zusammenhang mit allgemeiner Simulation gesehen (siehe z.B. [Law00]), diese Ansätze können natürlich auch auf Multiagentensysteme angewendet werden. Dort wird meist ein Simulationslauf als Individuum eines Genetischen Algorithmus betrachtet oder es werden suchbasierte Verfahren eingesetzt, um in der Response-Surface der Bewertungsfunktion ein Optimum zu finden. Allerdings wird dabei kaum beachtet, dass ein Simulationslauf mehrere Stunden dauern kann und deswegen nur extrem wenige Läufe überhaupt möglich sind. Mit geeigneten Heuristiken könnten diese Verfahren aber auch sinnvoll sein, da so nur wenige Simulationsläufe überhaupt durchzuführen sind oder Läufe frühzeitig durch eine Bewertungsfunktion abgebrochen werden können.

Optimierung in Multiagentensystemen kann allgemein auf verschiedenen Ebenen und durch unterschiedliche Methoden geschehen. Zum einen kann das Modell durch statische Analyse optimiert werden. Eine andere – schon vorgestellte – Möglichkeit ist es, einen ganzen Simulationslauf als parametrisierte Blackbox mit Ein- und Ausgabeverhalten zu

verstehen und das Optimierungsproblem mit Standardmethoden aus der Funktionsoptimierung zu lösen.

Zwischen diesen beiden steht das biologische Vorbild der Evolution, d.h. einzelne Individuen optimieren sich innerhalb eines Simulationslaufes. Dieses Verfahren hat den Vorteil, dass es nicht so zeitaufwendig wie die Blackbox-Methode ist und dass es ohne zusätzliches Wissen auskommt, welches bei analytischen Verfahren oft notwendig ist. Außerdem können mit diesem Verfahren biologische Fragestellungen bzgl. Evolution simuliert und bewertet werden.

Im nächsten Abschnitt wird nun diskutiert, welche bekannten Verfahren in unserem Zusammenhang sinnvoll sind, daran schließt sich eine Beschreibung des implementierten Verfahrens an. Zuletzt wird ein modelliertes biologisches Szenario vorgestellt und evaluiert. Im Rahmen eines Papiers muss natürlich an einigen Stellen auf große Tiefe verzichtet werden, der interessierte Leser sei allerdings auf [Hör00] verwiesen.

## **Was wird evolviert**

Zuerst ist zu klären, was evolvierbar sein kann und was evolviert werden soll. In unserer Simulationsshell SeSAM wird das Verhalten der Agenten durch eine endliche Maschine bestimmt, die aus Aktivitäten (Zuständen), Übergangsregeln und Aktionen (die innerhalb einer Aktivität ausgeführt werden) besteht. Für eine genaue Beschreibung von SeSAM siehe z.B. [Klü98].

Eine mögliche Vorgehensweise ist, der Evolution die Veränderung des gesamten Verhaltens zu ermöglichen, d.h. weder Art noch Anzahl von Zuständen, Regeln und Aktionen wäre konstant. Damit wäre diese Vorgehensweise ähnlich zum *Genetischen Programmieren* von John Koza, [Koz99], bei dem die Mutation an den Parse-Bäumen von Programmen (fast) beliebige Knoten- und Kanten-Änderungen vornehmen kann und die Rekombination die Parse-Bäume mehrerer Programme durch Austauschen von Teilbäumen mischt. Dadurch können ‚beliebig‘ komplexe Programme erzeugt werden, die ein gegebenes Problem lösen.

Eine andere Vorgehensweise ist, nur die Veränderung von numerischen Terminalen zuzulassen - ähnlich der einfacheren *Genetischen Algorithmen* von John Holland [Hol75] oder den *Evolutionsstrategien* von Ingo Rechenberg [Rech73] und Hans-Paul Schwefel [Schwe75]. Dabei wäre die Anzahl aller Regeln, Zustände und Aktionen vorgegeben und die Veränderungen derart eingeschränkt, dass zwar einige Terminale geändert werden können, die Länge und Struktur der Regeln und Aktionen festgelegt ist. Es ist zwar durchaus denkbar, den gesamten Aktivitätsgraphen in einem Bitstring zu kodieren, allerdings würden dadurch viele Vorteile eines GAs verloren gehen.

Der Nachteil einer deswegen begrenzten Evolution ist oft und auch im Rahmen unserer Arbeit ein Vorteil: Nur durch Beschränkungen ist es möglich, biologische Evolution *effizient* zu simulieren.

Bei Verwendung der GP-Operatoren wäre es der Evolution möglich, Verhalten zu erschaffen, das von enormer Komplexität ist. In den meisten Fällen ist jedoch dieses Verhalten sinnlos und würde nur zu höherer Rechenzeit führt, in anderen Fällen kann es zwar auch besonders effektiv sein - aber auch besser als das beste Verhalten, das der Modellbauer vorgesehen hat. Dieses Problem könnte durch komplizierte Testmechanismen beseitigt werden, allerdings nur auf Kosten der Effizienz.

Beim einfacheren GA- bzw. ES-Ansatz genügt es, Verhalten innerhalb der gewünschten Komplexitätsgrenzen einzugeben und der Evolution das ‚Fein-Tuning‘ dieses Verhaltens zu überlassen.

Ein weiteres Argument für den zweiten Ansatz liegt in der zugrundeliegenden Simulationsschale SeSAM begründet: Eine Integration von GP ist zwar theoretisch möglich, allerdings verhindert die momentane Implementation von SeSAM dieses Vorhaben.

## **Wie wird evolviert**

Sowohl für die GA (erweitert auf reelle Zahlen) als auch für ES wurden genetische Operatoren und Selektions- und Heiratsschemata entwickelt, die ohne größere Änderungen verwendbar sind. Die Schemata sind aber bei der Nachbildung von biologischer Evolution eigentlich nicht notwendig, da die Selektion durch ‚survival of the fittest‘ und Heirat durch räumliches Aufeinandertreffen von Agenten simuliert wird.

Für eine möglichst genaue Nachahmung biologischer Evolution ist es nicht notwendig, das Erbgut als Strang von Basen zu simulieren, denn die genauen Mechanismen auf Molekülebene sind nicht genügend bekannt, um exakt nachbildbar zu sein. Als weiteren wichtigeren Punkt würde dieser Mechanismus zusätzliche Rechenzeit erfordern, wobei momentan keine Vorteile dieser Abbildung gegenüber einer direkten Gen-Merkmal Abbildung bekannt sind.

Ein Kennzeichen der natürlichen Evolution ist, dass kleine Mutationen wahrscheinlicher sind als große. Dies liegt daran, dass starke, sinnvolle (d.h. nicht-letale) Mutationen nur durch Änderungen vieler Gene bzw. Basenpaare möglich ist, das dazu erforderliche genau gleichzeitige Auftreten dieser Veränderungen ist aber auch unwahrscheinlicher als die für kleine Mutationen notwendigen gleichzeitigen Änderungen weniger Gene. Allerdings können sich durch Abhängigkeiten zwischen den Genen Wirkketten bilden, die dazu führen können, dass auch kleine Änderungen große Auswirkungen haben.

Die Gene eines Agenten repräsentieren numerische Variablen, die im Verhalten des Agenten (mehrfach) verwendet werden können. Damit stellt ein solches Gen kein biologisches Gen dar, sondern das Ergebnis des Zusammenwirkens vieler Gene. Folglich müssen bei der Mutation eines Agent-Genes die erwähnten Wahrscheinlichkeiten berücksichtigt werden.

Prinzipiell lässt sich aber die Mutation auch nach anderen Verteilungen durchführen, z.B. nach einer Gleichverteilung. Dann wird zwar der Mutationsteil der Evolution zu einer Zufallssuche, aber man kann die Anwendung der Gleichverteilung auf einzelne Gene beschränken - auch freie Verteilungen mit beliebig eingebbaren Dichtekurven sind möglich.

Allerdings führen ineinander verschränkte Genwirkketten auch zu einem Effekt, der sich nachzubilden lohnt: Viele Lebewesen besitzen anstatt haploidem Erbgut diploides Erbgut, d.h. jedes Merkmal wird im Erbgut von zwei Allelen repräsentiert, allerdings werden nicht beide Allele exprimiert, sondern nur das dominanteste (bei dominant/rezessiven Erbgängen) oder eine Mischung der beiden (bei intermediären Erbgängen).

Dadurch kann sich ein genetisches Gedächtnis bilden, da nur die exprimierten Allele von der Evolution bewertet werden, die nicht-exprimierten können sich dagegen beliebig oder gar nicht verändern - erst bei einem (seltenen) Dominanzwechsel ‚kippt‘ die Merk-

malsausprägung und der früher gute Phänotyp (d.h. Summe aller Merkmale) kommt wieder zum Vorschein.

Das polyploide Agenten-Genom wird deswegen in zwei Arten von Genen aufgeteilt: Die Gene der einen Art repräsentieren die exprimierten Merkmale und die der anderen für die zur Wahl stehenden Allele. Die Bestimmung der Merkmalsausprägung aus den Allelen kann dann entweder dominant/rezessiv oder intermediär geschehen.

Bei einem Dominanzwechsel, dessen Wahrscheinlichkeit zu Beginn der Simulation festgelegt wird, ermittelt die Evolutionskomponente dann entweder für alle Allelgruppen eine neue einheitliche Dominanzreihenfolge oder jede Allelgruppe wird nach einer eigenen Reihenfolge neu geordnet. Die unterschiedlichen Auswirkungen lassen sich so zusammenfassen: Bei der ersten Möglichkeit merkt sich das genetische Gedächtnis immer einen kompletten Phänotyp, im zweiten Fall höchstens einzelne Werte.

Ein Agent-Gen besitzt folgende Attribute:

- Identifikator
- Wertebereich
- aktueller Wert
- Verteilung
- aktuelle Standardabweichung
- Dominanzwert

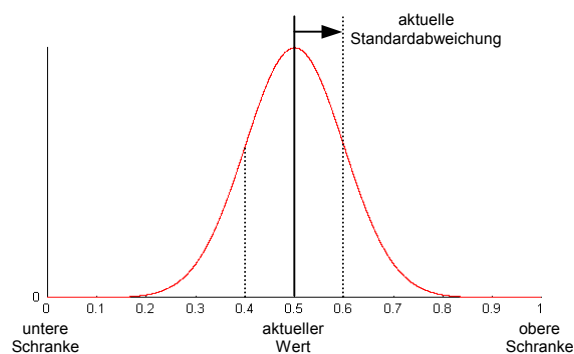


Abbildung 1: Gen

Die Einschränkung des Gens auf einen Wertebereich ist zwar prinzipiell nicht nötig, erspart aber eine Anpassung für sinnlose Werte (negative Wahrscheinlichkeiten, etc.).

### Rekombination

Auch für die Rekombination existieren diverse Operatoren - weil aber jedes Gen individuelle Wertebereiche und Verteilungen besitzen kann, müssen die eingesetzten Operatoren auf solche beschränkt werden, deren Anwendung immer Genome erzeugt, die jedes Gen (genau) einmal besitzen, wie z.B. „partially matched crossover“, „uniform oder based crossover“ und „ordered crossover“. Am biologisch sinnvollsten ist es,

- das eine Genom entsprechend des anderen zu ordnen
- und einen n-point-crossover durchzuführen.

Bei polyploiden Genomen wird das Genom in haploide Stränge aufgespalten und diese werden miteinander rekombiniert - anschließend werden die einzelnen Stränge wieder zu einem polyploiden Genom verschmolzen.

### Experimente

Die Evolutionskomponente wurde anhand ihrer Anwendung in drei verschiedenen Szenarios entwickelt, d.h. die volle Funktionalität stand erst beim letzten Szenario zur Verfüg-

gung, in dem die Evolution von solitär lebenden zu staatenbildenden Insekten nachvollziehbar werden sollte.

Das Verhalten der Insekten ist dabei in vier Hauptbereiche aufgeteilt:

- Standardverhalten, das allen Insekten gemeinsam ist: Wachsen, Schlüpfen, Nahrung im Nest aufnehmen
- Jagd-Verhalten: Beute suchen, erlegen und als Nahrung im Nest einlagern, Angreifer abwehren
- Brutpflege-Verhalten: Brut mit Nahrung aus dem Nest oder aus dem eigenen Magen versorgen
- Königin-Verhalten: Paarung, Eier legen, Verfliegen (d.h. neue Nistplätze beziehen)

In der Natur gehen Änderungen des Verhaltens immer einher mit Veränderungen des Körperbaus - diese werden im Modell allerdings nicht ermöglicht bzw. berücksichtigt. Stattdessen wird die Aktivierung einfach als Königin-, Jagd- und Brutpflege-Faktor vererbt, d.h. je höher der Faktor, desto eher wird das Insekt das entsprechende Verhalten ausführen. Je höher die verschiedenen Faktoren, desto höher werden aber auch die Kosten für die Produktion eines solchen Individuums: Die Nahrungsmenge, die einem solchen Individuum während der Brutzeit gefüttert werden muss, ist proportional zur Summe aller Faktoren (wobei der Königin-Faktor schwerer wiegt). Damit soll erreicht werden, dass eine Individuum (bzw. die Evolution) abwägen muss: Entweder viele einfache, spezialisiere Individuen oder wenige Allrounder, d.h. eigentlich ‚normale‘ Individuen.

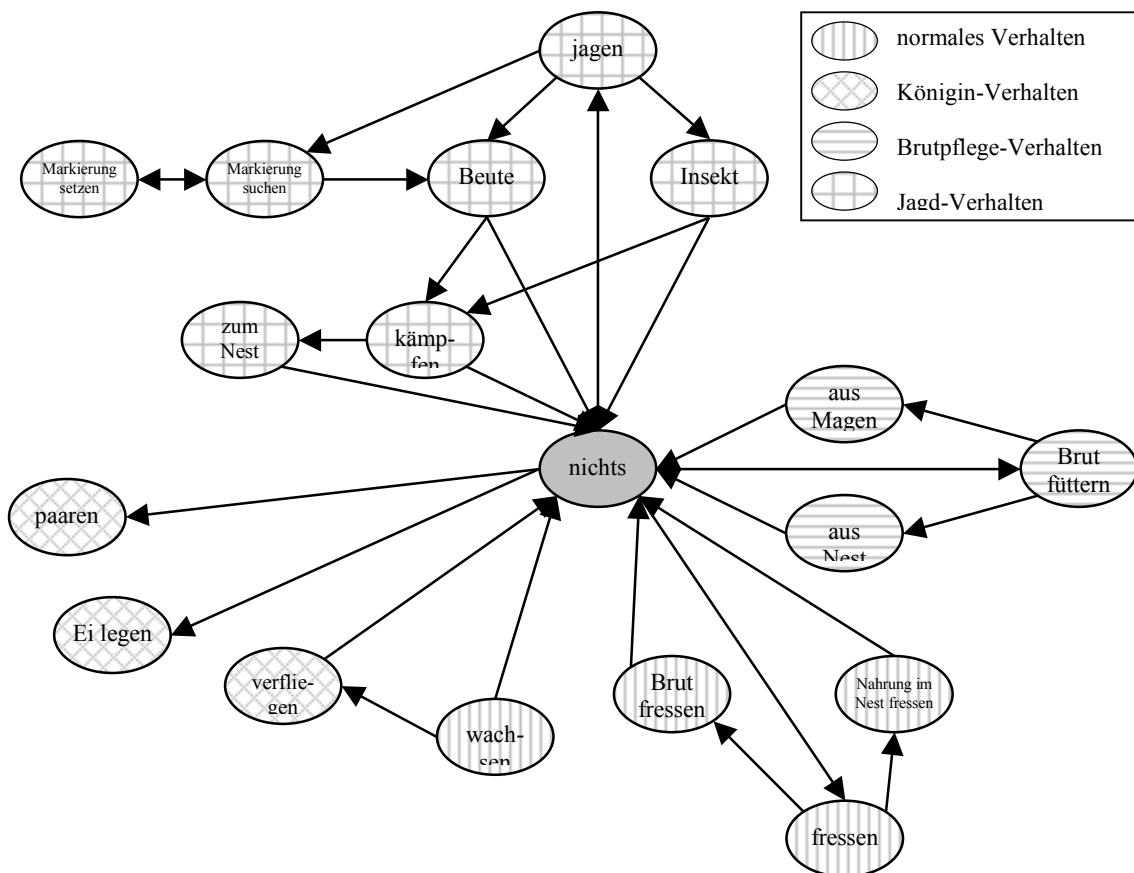


Abbildung 1: Insektenverhalten

In Abbildung 2 ist der gesamte Aktivitätsgraph eines Insekts aufgetragen, dabei wurden die Übergangsmöglichkeiten durch gerichtete Pfeile gekennzeichnet, die genauen Übergangsregeln (einschl. Wahrscheinlichkeiten) wurden allerdings vernachlässigt.

Insgesamt besitzt ein Insekt 13 Gene (bei haploidem Genom), die folgendermaßen verwendet werden:

Gen	Wirkung
Faktor_Königin	Motivation für Königin-Verhalten (bei Unterschreitung der simulationsspezifischen Königin_Schranke wird der Wert bei der Ausprägung auf 0 gesetzt)
Faktor_Jagd	Motivation für Jagd-Verhalten, Tauglichkeit im Kampf mit anderen Insekten
Faktor_Brutflege	Motivation für Brutpflege-Verhalten
Energie_minimal Energie_maximal	Unterer und oberer Level des Energiehaushalts. Die beiden Level legen fest, in welchem Energierahmen das Insekt versucht zu leben. Je niedriger der untere Level, desto schneller stirbt das Insekt bei Unterschreitung. $p(\text{Insekt stirbt}) = \frac{\text{Energie_minimal} - \text{aktuelle_Energie}}{\text{Energie_minimal}}$
Energie_abliefern_Faktor	Wenn ein Insekt Nahrung im Nest einlagert, dann liefert es $(\text{aktuelle_Energie} - \text{Energie_minimal}) * \text{Energie_abliefern_Faktor}$ ab.
Energieportion_Ameise	Pro Fressvorgang entnimmt die Ameise dem Nest Energie in Höhe von Energieportion_Ameise
Brut_fressen_Faktor	Wenn sich im Nest keine Nahrung aber noch Brut befindet, dann frisst das Insekt die Brut mit der geringsten Energie, wenn $\frac{\text{aktuelle_Energie} - \text{Energie_minimal}}{\text{Energie_maximal} - \text{Energie_minimal}} < \text{Brut_fressen_Faktor}$
Energieportion_Brut	Pro Brutpflegevorgang entnimmt die Ameise dem Nest oder ihrem Magen Energie in Höhe von Energieportion_Brut
Energie_für_Ei	Wenn eine Königin ein Ei legt, dann gibt sie dem Ei einen Teil ihrer eigenen Energie als ‚Startkapital‘
Eier_minimal Eier_maximal Eier_pro_Paarung	Wenn die Anzahl der befruchteten Eier eines Insekts unter Eier_minimal fällt, dann versucht es sich mit zunehmender Wahrscheinlichkeit zu paaren. Dabei können aber insgesamt höchstens Eier_maximal Eier befruchtet werden. Pro Paarungsvorgang wird eine Anzahl Eier gemäß Eier_pro_Paarung befruchtet.

**Tabelle 1: Insektengene**

Zusätzlich zu diesen ererbten Parametern besitzt ein Insekt noch weitere, die seinen aktuellen Zustand bestimmen wie Energie, Alter, Wahrnehmungsradius, Geschwindigkeit, etc.. Zusätzlich besitzen die Insekten die Möglichkeit, ihre gegenseitige Verwandtschaft festzustellen. Diese Verwandtschaft bestimmt, ob sich Insekten gegenseitig bekämpfen oder nicht.

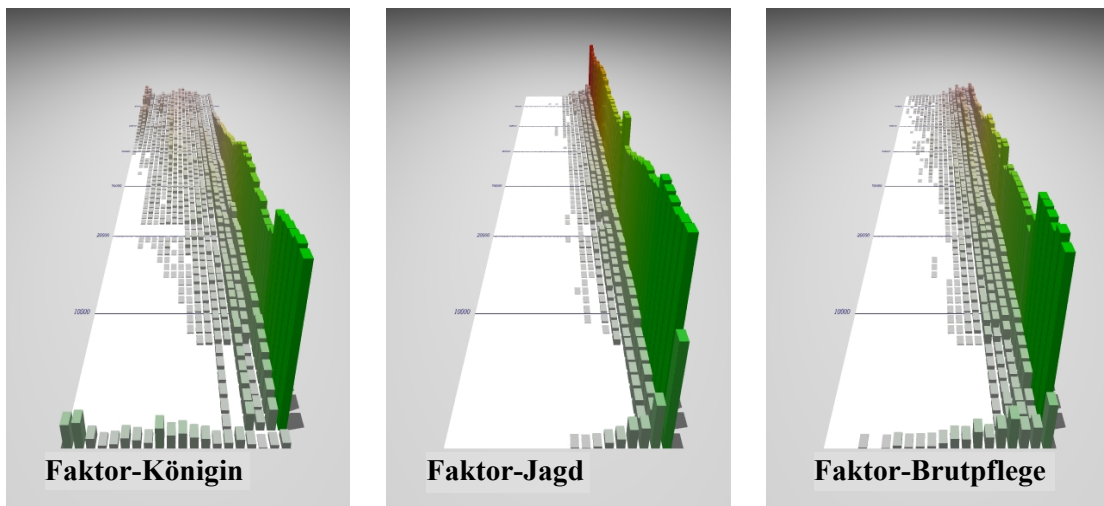
## Resultate

Die Simulationen wurden mit einer 200x140 Felder großen Welt durchgeführt, auf der durchschnittlich 1500 Beuteobjekte (die zufällig erzeugt wurden und nach einiger Zeit auch wieder verschwanden) und ca. 60 Nistplätze mit unterschiedlichem Abstand zueinander existierten. Zu Beginn der Simulation wurden ungefähr die Hälfte der Nistplätze mit einzelnen Insekten besetzt.

Pro Takt kann ein Insekt alle Aktionen einer Aktivität durchführen, also:

- Bewegung um maximal ein Feld
- Fokussierung, d.h. seinen Fokus auf ein anderes Objekt ändern
- Energie transferieren
- Kämpfen
- Paarung vollziehen
- Ei legen, d.h. Insektenlarve erzeugen
- Larvenstadium verlassen

Die Simulationen zeigten nach ungefähr 12000 Takten (mit Königin\_Schranke = 0.1, siehe Tabelle 1) folgende Ergebnisse:



**Abbildung 2: Veränderungen des Genpools**

**Der Wertebereich des Gens ist entlang der x-Achse aufgetragen, die Höhe der Balken gibt die Häufigkeit des entsprechenden Wertebereich-Ausschnitts an und entlang der z-Achse ist die zeitliche Veränderung zu erkennen. Die erste Reihe zeigt jeweils das Ergebnis am Ende der Simulation.**

Man kann erkennen, dass sich der Faktor-Königin fast bis auf 0 ändert (und zwar so, dass ca. 10% aller Nachkommen Königin-Verhalten ausführen können), wohingegen sich die Faktoren Jagd und Brutpflege kaum verändern. Eine genauere Analyse zeigte folgendes: Je weiter die Nester voneinander entfernt sind, desto eher darf der Faktor-Jagd abnehmen, da die Beuteobjekte beim Kampf immer verlieren und der Kampf gegen andere Insekten seltener wird. Auch bei besonders viel Beuteressourcen fällt der Jagd-Faktor, da immer bevorzugt Beuteressourcen als Nahrung verwendet werden. Der Faktor-Brutpflege fällt dann, wenn viel Nahrung vorhanden ist und der Wert für Energieportion-Brut steigt, da

bei großen Portionen nicht ständig gefüttert werden muss. Im Zuge der entstehenden Staatenbildung (d.h. Insekten, die kein Königin-Verhalten ausführen können, werden produziert) fällt auch immer der Wert von `Energie_für_Ei`: Da ja dann Insekten vorhanden sind, die nur Brut pflegen und jagen können, ist es für die Königin nicht mehr nötig, mehr Energie als unbedingt nötig in die Produktion von Nachkommen zu investieren - dafür, dass keine Brut verhungert, sind ja inzwischen Brutpflegerinnen vorhanden. Ein ebenfalls interessantes Ergebnis ist die bei allen Simulationen aufgetretene Nicht-Steigerung von `Eier_pro_Paarung`: Anscheinend ist es nicht günstig, zu viele Eier zu befruchten, da sonst beim Legen der letzten Eier schon zu veraltetes Verhalten verwendet werden würde.

Dieses Experiment zeigt also, dass es unter bestimmten Umständen für einzelne Individuen sinnvoller ist, die eigene Fortpflanzung als weniger wichtig zu erachten und statt dessen für den Staat zu arbeiten. Mit weiteren Experimenten und der Einbeziehung von Biologen soll zum einen das Modell validiert und die Ergebnisse besser beurteilt werden.

## **Zusammenfassung**

In diesem Papier wurde eine Komponente für SeSAM vorgestellt, mit deren Hilfe Evolution in einem Multiagentensystem nachgebildet werden kann. Der evolutionäre Algorithmus ist dabei sehr ähnlich zu einer Evolutionsstrategie, es können nur Parameter eines Agenten weitervererbt und verändert werden. Diese Komponente wurde bis jetzt in drei Anwendungen evaluiert und es zeigte sich, dass es dem Modellbauer damit möglich ist, komplexe Szenarien zu entwerfen und aus den Ergebnissen aussagekräftige Schlüsse zu ziehen. Hierbei sei vor allem auf die positiven Ergebnisse einer Evolution von solitär lebenden hin zu staatenbildenden Insekten hingewiesen, die in diesem Papier beschrieben ist.

Wie erwartet, ist ein großes Problem die Effizienz. Sie kann durch ein geeignetes Modell mit möglichst wenigen, aber trotzdem genügend vielen, vererbbaaren Parametern gesteigert werden. Eine Parallelisierung der Simulation ist schon dadurch gegeben, dass zahlreiche<sup>1</sup>, gleiche Simulationsläufe auf unterschiedlichen Rechnern ablaufen können. Zu überlegen ist noch, ob eine Verteilung eines Simulationslaufes auf mehrere Rechner sinnvoll ist. Eine naheliegende Aufteilung der Agenten auf die verschiedenen Rechner wäre durch ihre räumliche Beziehung gegeben.

Bei diesen ganzen Untersuchungen darf nicht vergessen werden, dass durch diese Evolution das Modell optimiert wird. D.h. durch die vorgestellte Evolutionskomponente ist eine automatische Optimierung eines Modells möglich. In diesem Zusammenhang muss noch untersucht werden, wie eine Bewertungsfunktion in die vorgestellte Evolutionskomponente eingearbeitet werden kann. Momentan wird rein biologisch nach ‚survival of the fittest‘ selektiert. Eine Möglichkeit wäre es nach  $n$  Takten gemäß eines normalen steady-state GA alte Individuen durch neue zu ersetzen, wobei dann die Bewertungsfunktion mit einbezogen wird. Andererseits könnte auch im Modell eine Präferenz hin zu höherbewerteten Individuen bei der Partnerauswahl berücksichtigt und schwachbewertete Individuen von der Umgebung aus dem Genpool entfernt werden.

---

<sup>1</sup> Zahlreiche deswegen, da die Evolutionskomponente (und das Modell) nichtdeterministisch ist.

## **Literaturverzeichnis**

[Hör00] Hörnlein: „*Ethologisch motivierte evolvierbare Agentenarchitektur*“, Diplomarbeit, Universität Würzburg, 2000

[Hol75] Holland: „*Adaptation in natural and artificial systems*“, University of Michigan Press, Ann Arbor, Michigan, 1975.

[Klü98] Klügl, Puppe: „*The Multi-Agent Simulation Environment SeSAM*“, In: H. Kleine Büning (Hrsg.): Proceedings des Workshops "*Simulation in Knowledge-based Systems*", Paderborn, April 1998 (= Report tr-ri-98-194, Reihe Informatik, Universität Paderborn)

[Koz99] Koza: „*Genetic Programming III*“, Morgan Kaufmann Publishers, Inc., San Francisco, 1999.

[Law00] Law, Kelton: „*Simulation Modelling and Analysis*“; 3rd Edition; 2000

[Oechs99] Oechslein, Klügl, Puppe: „*Kalibrierung von Multiagentenmodellen*“, 13. Workshop der ASIM-Fachgruppe "*Simulation und Künstliche Intelligenz*", In Chemnitzer Informatik-Berichten (CSR-99-03)

[Rech73] Rechenberg: „*Evolutionsstrategie*“, Frommann-Holzboog, Stuttgart, 1973

[Schwe75] Schwefel: „*Dissertation: Evolutionsstrategie und numerische Optimierung*“, Technische Universität Berlin, Mai 1975