

Evolution und Lernen: individuelle Adaption und Evolution bei simulierten Agenten

Tobias Jung
15.05.2001

Zusammenfassung

Dieser Artikel gewährt einen Überblick über meine Diplomarbeit, die Auswirkungen untersucht, welche bei der Interaktion von evolvierten Sensoren mit der Fähigkeit zur individuellen Adaption bei simulierten autonomen Agenten auftreten können. Dabei erlaubt es Evolution den Agenten, Sensorstrukturen und Steuerungsmechanismen simultan über einen großen Zeitraum und viele Generationen hinweg zu entwickeln, während es auf einer kleineren Zeitskala einem Agenten möglich ist, sein Aktionsmodul so zu verändern, daß er in seiner Umwelt am besten überleben kann.

Um ein überschaubares und transparentes Modell in möglichst allgemeiner Form zu konstruieren, wird die Umwelt als diskreter Markov-Entscheidungsprozeß (MDP) formuliert, und die Sensoren als abstrakte Abbildung zwischen Welt- und Agentenzustand implementiert. Ein genetischer Algorithmus treibt die Entwicklung der Sensoren auf Populationsebene voran, während der lokale Lernvorgang durch *Reinforcement Learning* Algorithmen (*Q-Learning*) realisiert wird. Neben simulationsbasierten Ergebnissen werden aktuelle informationstheoretische Maße und Methoden herangezogen, die ein weiteres Verständnis der zugrundeliegenden Mechanismen erlauben.

1 Einleitung

1.1 Sensor-Evolution

Die erfolgreichsten autonomen Agenten hat die Natur bislang selbst hervorgebracht: durch Evolution an ihre Umgebung angepaßt, sind sie in der Lage, allen Unzulänglichkeiten und kleinen "Überraschungen" einer dynamischen Umwelt zum Trotz zu überleben. Solchermaßen inspiriert, werden seit geraumer Zeit im Gebiet *Artificial-Life* evolutionäre Algorithmen nicht nur zur Entwicklung von Steuerungsmechanismen, sondern auch zum Design von Sensoren eingesetzt. Evolution von Sensoren kann dabei bedeuten, die Art (z.B. visuell, taktil), die Funktionsweise (z.B. Lichtwellen, Radar), einen charakteristischen Parameter (z.B. Schwingel) oder die Anzahl/Konfiguration zu variieren, siehe etwa [1, 7, 5].

In einer weniger am biologischen Vorbild orientierten als abstrakten Sichtweise werden hier Sensoren als Abbildung zwischen Weltzustand und internem Agentenzustand aufgefaßt, also als Schnittstelle von Agent und Umwelt; sie bestimmen die Qualität und Quantität von aufgenommenen Informationen und bestimmen so im Modell die Überlebenswahrscheinlichkeit bzw. die reproduktive Fitness.

1.2 Zielsetzung der Diplomarbeit

Aus der Anwendung ist bekannt, daß evolutionäre Algorithmen auf sinnvolle Weise mit Lernen verbunden werden können; solche hybriden Ansätze bieten die Möglichkeit, einen größeren Teils des Lösungsraums zu erforschen, als jedes Verfahren allein könnte, indem Evolution als globale Komponente auf Populationsebene und Lernen als lokale Suche auf der Ebene des Individuums wirken. Zielsetzung der Diplomarbeit ist es nun zu untersuchen, ob die Lernfähigkeit eines Individuums die Evolution begünstigen kann. Der wesentliche Unterschied zum klassischen *Baldwin-Effekt* [2], der besagt, daß es Formbarkeit auf Ebene des Individuums ("*phenotypic-plasticity*") erlaubt, erworbene Charakteristika indirekt zu vererben, besteht in einer strengen Trennung von Evolution und Lernen, indem beide auf unabhängig modellierten Räumen operieren.

Um die Auswirkung dieser Interaktion zu untersuchen, muß ein überschaubares und transparentes Modell zur abstrakten Sensor-Evolution konstruiert werden, das es erlaubt, die Mechanismen zu identifizieren, die bei Adaption unterschiedlicher Intensität zu neuen Generationen von Sensoren führen. Sowohl ein qualitatives Maß für die Überlebensfähigkeit des Agenten in der Umwelt, als auch ein informationsbasiertes auf dem Raum der Sensorlösungen sind bei der Untersuchung verwendet worden.

2 Das Szenario

2.1 Übersicht

Das Szenario besteht aus einer Population evolvierbarer Agenten, die mit einer einfachen Umgebung interagieren. Um Interaktion zwischen Agenten untereinander zu vermeiden, ist die Aufgabe so gestaltet, daß sich jeder Agent in einer eigenen Umwelt befindet.

Die Agenten sind reaktiv modelliert: zu diskreten Zeitpunkten empfangen sie Signale aus der Umwelt und reagieren auf diese Stimuli mit passenden Aktionen. Für die Komponente,

die aus Signalen passende Aktionen generiert, wird im weiteren der Begriff Aktionsmodul austauschbar mit Kontrollmodul verwendet.

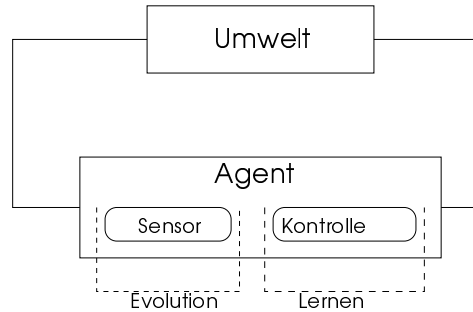


Abbildung 1: Struktur des Modells

Die Architektur der Agenten setzt sich aus getrenntem Sensor- und Aktionsmodul zusammen (s. Abb. 1):

1. Ein genetisch festgelegter Sensor bildet globale Weltzustände in eine Menge möglicher Beobachtungen des Agenten, seine begrenzte Sicht auf die Welt, ab.
2. Das Kontrollmodul wählt abhängig vom Sensorsignal eine Aktion aus einer Menge von möglichen aus. Diese Abbildung wird für jeden Agenten individuell durch Q-Learning [11] adaptiert.

Das Ziel der evolutionären Suche ist es, die Sensorabbildung derart zu optimieren, daß sie *zusammen* mit dem Aktionsmodul eine gute Leistung erzielt. Während die Sensoren im genetischen Bauplan verankert sind, ist das Aktionsmodul zu Beginn der Lebenszeit bei allen Agenten gleich gewählt. Die aus einem Genom resultierenden Phänotypen (Agenten) werden einzeln in der Umwelt evaluiert, indem der Agent versucht, durch Lernen sein Kontrollmodul so zu verändern, daß seine Fitness maximiert wird. Die simulierte Umwelt ist eine Gitterwelt, die sich sehr elegant als abstrakter Entscheidungsprozeß auffassen lassen kann.

2.2 Ein Gitterweltszenario

Als Aufgabe wurde für jedes Individuum ein einfaches Räuber-Beute-Szenario gewählt, wobei die Agenten nur die Rolle des Räubers übernehmen, die Beute wird als fester Teil der Umwelt angesehen und folgt einer simplen¹ Ausweichstrategie.

Die Umwelt ist eine toroide Gitterwelt mit 11x11 Feldern, Abb. 2 zeigt eine typische Situation und den Zusammenhang mit dem dadurch repräsentierten Zustand.

Die Fitness, die in dem evolutionären Prozess die Selektion bestimmt, und daher das Ziel des Agenten, ist sich in möglichst wenigen Schritten auf das Feld der Beute zu bewegen, d.h. ein Modell (implizit) für die Bewegung der Beute zu lernen. Agent und Beute bewegen sich sequentiell; der Agent besitzt eine Orientierung und kann je Simulationsschritt eine von 6

¹die Ausweichstrategie berücksichtigt nur die vorhergehende Aktion des Räubers und stellt so sicher, daß die Markoveigenschaft des Prozesses erhalten bleibt, d.h. ein neuer Zustand nur von dem unmittelbar vorhergehenden abhängt.

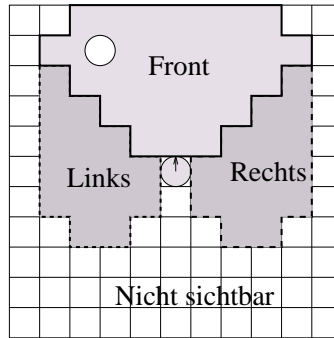


Abbildung 2: Zusammenhang zwischen Umwelt und Zustand des Systems. Der graue Kreis in der Mitte repräsentiert den Agenten mit Blickrichtung, der weiße Kreis die Beute. Die relative Position der Beute bestimmt den Zustand des Systems. Die schraffierten Bereiche markieren als Front- und Seitenbereich das Blickfeld des Agenten.

Aktionen durchführen (s. Abb. 3), sofern diese Bewegung nicht auf dem Feld der Beute geendet hat, erfolgt der Zug der Beute. Das Problem ist episodischer Natur: war der Zug erfolgreich (für den Agenten), wird ein neues Beute-Individuum generiert und eine neue Episode beginnt mit zufälligen Startpositionen. Während der Lebensdauer eines Agenten werden normalerweise viele solcher Episoden durchlaufen.

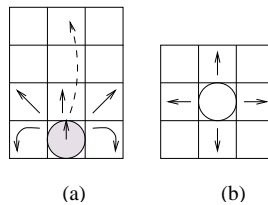


Abbildung 3: (a) Die 6 möglichen Aktionen des Agenten. (b) Die möglichen Züge der Beute.

Dies ist eine typische Aufgabe für RL, die Belohnung für Aktionen wird erst verzögert, mit dem Ende der Episode sichtbar. Die Umgebung ist deterministisch, stationär und diskret.

2.3 Abstrakte Sensoren

Von einem abstrakten Standpunkt betrachtet, kann ein Sensor als Abbildung von Zuständen der Umwelt in eine interne Zustandsmenge – Menge von Beobachtungen – eines Individuums aufgefaßt werden. Im obigen Szenario ist ein Zustand durch die relative Position der Beute zum Agenten bestimmt, die globale Umwelt nimmt also einen von 121 verschiedenen Zuständen an.

Der Agent ist mit einem eingeschränkten Blickfeld (s. Abb. 2) modelliert, so daß er nur 63 Felder einsehen kann. Der Zustandsraum² für den Agenten kann demzufolge als $\mathcal{S} = \{s_0, \dots, s_{63}\}$ geschrieben werden, wobei s_{63} ein künstlicher Zustand ist, der vorliegt,

²Das Blickfeld wird im Modell von getrennten Front- und Seitensensoren erfaßt. Da aber immer nur ein Feld "aktiv" ist, können die möglichen Zustände durchnummeriert werden.

wenn die Beute keines der 63 sichtbaren Felder einnimmt. Analog wird eine Menge interner Zustände bzw. möglicher Beobachtungen $\mathcal{X} = \{x_0, \dots, x_{63}\}$ gewählt.

Eine elementweise Abbildung $\sigma : \mathcal{S} \rightarrow \mathcal{X}$ modelliert dann in abstrakter Weise einen Sensor. Somit existieren zwei Systeme: neben dem tatsächlichen Markov-Prozess auf \mathcal{S} , in dem die Auswirkungen der Aktionen des Agenten sichtbar werden, kann der Agent nur den Prozess in $Bild(\sigma) \subseteq \mathcal{X}$ verfolgen.

Diese Sensorabbildung ist der Evolution unterworfen, wobei folgendes gilt: außer der Identität $\sigma(s_k) = x_k$, $k = 0, 1, \dots, 64$ (und alle Permutationen davon), die die Umwelt vollständig erfaßt, ist σ nicht surjektiv, $Bild(\sigma) \subset \mathcal{X}$ (und damit auch nicht injektiv). Insbesondere existieren also Zustände $s_i \neq s_j$ mit $\sigma(s_i) = \sigma(s_j)$, die für den Agenten nicht unterschieden werden können.

Da die Möglichkeiten des Agenten, Aktionen auszuwählen, allein von $Bild(\sigma)$ abhängt, ist es wesentliche Aufgabe der Evolution, σ so zu wählen, daß die Dynamik des tatsächlichen Systems in möglichst idealer Weise wiedergegeben wird. Ideal muß nicht bedeuten, die Umwelt 1:1 wiederzugeben, eine sinnvolle Reduktion des Zustandsraum kann genauso gut sein, wie sich in Abschnitt 3 noch zeigen wird.

2.4 Der Genetische Algorithmus

Das Genom definiert (s. Abb. 4) nun jedes individuelle σ als String der Länge 64 von Indizes g_i , mit $\sigma(s_i) = x_{g_i}$, wobei die g_i Einschränkungen der einzelnen Sektoren Front, Links, Rechts berücksichtigen.

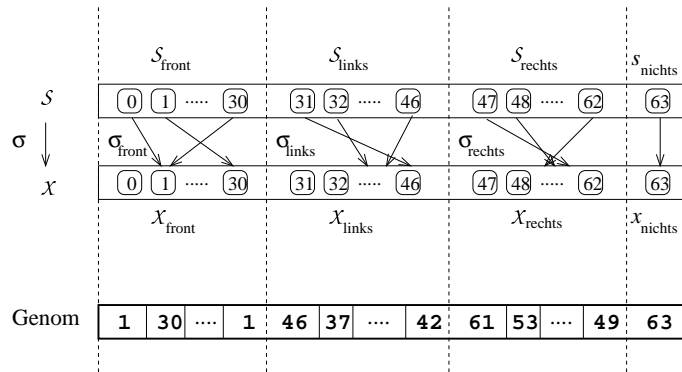


Abbildung 4: Die Sensorabbildung bildet die sichtbaren Felder \mathcal{S} in interne Zustände, die Menge der Beobachtungen \mathcal{X} , ab. Darunter die zugehörige Darstellung im Genom.

Zu Beginn eines Laufes besteht die Population aus zufällig initialisierten Genomen, Variation und Rekombination geschieht mit den genetische Operatoren Mutation und 1-Punkt Crossover .

2.5 Das Lernmodell

Mittels Q-Learning³ (1-Schritt Aktualisierung) werden die einzelnen Agenten auf \mathcal{X} eine Politik suchen, die die zum Erfolg benötigten Schritte minimiert (Belohnung $r = -1$ pro Schritt außer im Terminalzustand).

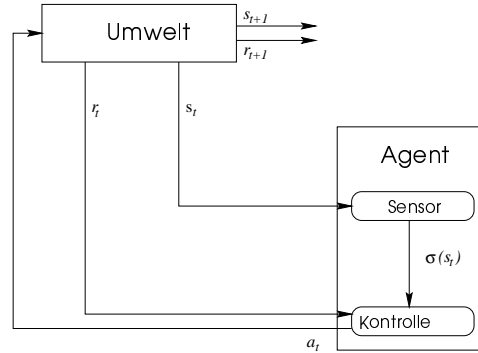


Abbildung 5: Interaktion von Umwelt und Agent

Im folgenden erläutere ich kurz, wie sich das Problem als Entscheidungsprozeß formulieren läßt, und Lernen ermöglicht wird.

Unser Entscheidungsprozeß besteht aus einem Agent und seiner Umwelt, die wie in Abb. 5 gezeigt, interagieren. Zu jedem Zeitpunkt t einer Folge diskreter Zeitschritte $t = 1, 2, 3, \dots$ nimmt der Agent den Zustand $s_t \in \mathcal{S}$ des Systems durch seine Sensorabbildung als Beobachtung $\sigma(s_t) = x_t \in \mathcal{X}$ wahr, und wählt Aktion a_t aus der Menge der möglichen Aktionen \mathcal{A} . Daraufhin wechselt die Umgebung ihren Zustand in einen neuen Zustand s_{t+1} und gibt dem Agenten für den Übergang von Zustand s_t nach s_{t+1} unter Aktion a_t die skalare Belohnung $r_{t+1} \in \mathbb{R}$ zurück. \mathcal{S} , \mathcal{X} und \mathcal{A} sind endlich. Das Ziel des Agenten ist es, die kumulierte Belohnung zu maximieren, die er auf lange Sicht erhalten wird, formal

$$E\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right)$$

Zum Finden einer Politik, die diesen Ausdruck für jeden Zustand maximiert, können *Temporal-Difference* Methoden verwendet werden, die ohne ein Modell für die Zustandsübergangswahrscheinlichkeiten zu besitzen, aus stochastischen Schätzungen, gewonnen aus tatsächlich durchgeführten Zustandsübergängen, eine Approximation der wahren Wertefunktion liefern. Die Q-Werte geben dabei bei diskreter Zustandsmenge in tabellarischer Form die Wertefunktion wieder, d.h. $Q(s, a)$ schätzt die zu erwartende Belohnung, wenn im Zustand s Aktion a ausgeführt wird. Die Lernregel für *1-step off-policy Q-Learning* lautet

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q_t(s_t, a_t) \right)$$

³tatsächlich ist das so formulierte Problem typischerweise nur ein partiell beobachtbarer Prozeß (POMDP, siehe etwa [3, 4, 10]) mit versteckten Zuständen. Insbesondere für die Theorie ist die Konvergenz von Q-Learning nicht mehr sichergestellt, gleichwohl es in der Praxis in diesem Kontext durchaus befriedigende Ergebnisse liefert.

wobei $0 \leq \gamma \leq 1$ der Diskontfaktor ist, der angibt, wie weit Schätzungen zukünftiger Belohnungen berücksichtigt werden sollen, $0 < \alpha \leq 1$ die Lernrate und Q_t die Schätzung der wahren Wertefunktion Q^* , zum Zeitpunkt t ist.

3 Informationstheorie und relevante Information

Da es der Aufgabe ein explizites Maß für die Güte von Sensoren mangelt, konnten wir bisher Aussagen über die evolvierten Sensoren nur anhand der in Simulationen erzielten Fitness treffen, bzw. diesen Wert zum Vergleich heranziehen. Es wäre aber wünschenswert, ein von der Simulation/Aufgabe unabhängiges Maß für die Sensoren zu finden, welches direkt die Güte beschreibt.

Der Nutzen von Sensoren hängt maßgeblich davon ab, inwieweit sie Informationen aus der Umwelt, die für das Erreichen einer hohen Fitness notwendig sind, übertragen können. Zur Quantifizierung dieser Information wurde für die Diplomarbeit ein Weg beschritten, der in Anlehnung an Shannons Entropiemaß [9] relevante Information im Sinne von Tishby et al. [12] und dessen konzeptuelle Erweiterung auf Entscheidungsprozesse von Polani et al. [8] vereinigt. Im folgenden werde ich dieses Konzept kurz beschreiben, wobei für eine detailliertere Diskussion auf [6] verwiesen wird.

3.1 Relevante Information

„Klassische“ Entropiemaße bieten bereits ein allgemeines Konzept zur Quantifizierung von Information, zeichnen sich aber gerade durch das Fehlen semantischer Informationen aus, in dem sie nur die Auftretenswahrscheinlichkeit der Signale, nicht aber ihre Bedeutung berücksichtigen. Insbesondere läßt sich Informationsverlust, der bei Kompression des Signalraums durch Sensoren auftreten kann, nicht befriedigend erklären.

Die Relevanz von Informationen einer Signalquelle kann nur zu einem gegebenen Kontext bestimmt werden. Im Hinblick auf den Entscheidungsprozeß erwächst die Bedeutung eines Zustands aus der Möglichkeit, eine Aktion zu wählen, die in diesem Kontext optimal ist, d.h. eine maximale Belohnung liefert. Dank *Q-Learning* sind wir in der Lage, zu jedem Zustand die optimalen Aktionen zu berechnen.

Daher kann Relevanz folgendermaßen definiert werden: die Umwelt, beschrieben durch den diskreten Zustandsraum \mathcal{S} , wird mit einer Zufallsvariable S assoziiert, mit Wahrscheinlichkeiten $p(s)$ für das Vorliegen von s . Analog beschreibt Zufallsvariable A die Auswahl einer optimalen Aktion, wobei diese Verteilung natürlich nicht unabhängig von S ist, sondern über die gemeinsame Verteilung $p(s, a) = p(s)p(a|s)$ mit

$$p(a|s) := \begin{cases} \frac{1}{|\mathcal{A}^*(s)|} & \text{falls } a \in \mathcal{A}^*(s) \\ 0 & \text{sonst} \end{cases}$$

definiert ist, wenn $\mathcal{A}^*(s)$ die Menge der in s optimalen Aktionen bezeichnet. Die relevante Information wird dann als Korrelationsentropie

$$I(S : A) := H(S) - H(S|A) = H(A) - H(A|S)$$

zwischen S und A definiert, wobei $H(S)$ die Entropie von S und $H(S|A)$ die bedingte Entropie ist. Anschaulich sagt diese Größe aus, wieviel Unsicherheit über die Wahl einer optimalen Aktion entfernt werden kann, wenn der Zustand bekannt ist (wegen Symmetrie gilt auch die Umkehrung, s. Abb. 6). Umgekehrt mißt $H(S|A)$ die mittlere Unsicherheit über den vorliegenden Zustand gegeben eine optimale Aktion; da die Kenntnis des genauen Zustandes allein für die Auswahl einer optimalen Aktion unerheblich ist, wird so für diesen Term der Begriff *irrelevante Information* motiviert.

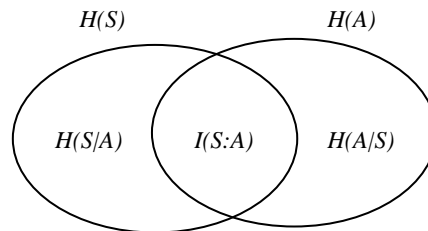


Abbildung 6: Darstellung der Korrelationsentropie als Venn Diagramm.

3.2 Relevante Information als Güte von Sensoren

In diesem Rahmen ist es nun quantitativ möglich, die Frage zu klären, wieviel relevante Informationen ein evolvierter Sensor übermitteln kann.

Der Zustandsraum wird durch Sensor σ in eine Teilmenge von Beobachtungen \mathcal{X} abgebildet, d.h. σ komprimiert den Zustandsraum in eine Darstellung, die (hoffentlich) möglichst viele Informationen beibehält. Die Verteilung der zugehörigen Zufallsvariable X kann, da der Sensor jedes $s \in \mathcal{S}$ deterministisch auf eine Beobachtung $x \in \mathcal{X}$ abbildet, einfach aus der totalen Wahrscheinlichkeit

$$p(x) = \sum_{s \in \mathcal{S}} p(s)p(x|s) = \sum_{s \in \mathcal{S}} p(s)1_{\{x=\sigma(s)\}}$$

berechnet werden. Die relevante Information $I(X : A)$ von X über A läßt sich nun einfach bestimmen, wobei nach Konstruktion $I(X : A) \leq I(S : A)$ gilt, d.h. die intuitive Vorgabe erfüllt, daß ein Sensor bestenfalls genausoviel Informationen transportieren kann. Daneben beschreibt $H(X|A)$ die Menge der irrelevanten Informationen, die ein Sensor beibehält.

Es ist möglich, einen Sensor zu konstruieren, der die gesamte Entropie $H(X)$ solchermaßen reduziert, daß er alle relevanten Informationen beibehält, aber stark den irrelevanten Anteil reduziert.

4 Simulationen

4.1 Implementierung

Die Implementierung der Simulationsumgebung geschah unter Verwendung der Programmiersprache C++. Durch streng objektorientiertes Programmdesign der essentiellen Programmteile konnte sowohl eine zweckmäßige Konsolenapplikation unter Solaris, als auch

eine GUI-Version unter Windows mit VisualC++ realisiert werden. Der genetische Algorithmus wurde unter Zuhilfenahme und Erweiterung der Objektbibliothek GALIB [13] implementiert. Für den genetischen Algorithmus wurde ein *steady state* mit Populationsgröße 20, Austausch 65% pro Generation, Mutationsrate 5% und Crossoverrate 70% verwendet. Ein typischer Simulationslauf mit 5000 Generationen benötigt im Mittel 6 Stunden.

4.2 Qualitative Charakterisierung evolvierter Sensoren

Es wurden nun verschiedene Simulationsläufe mit dem Ziel durchgeführt, den Einfluß unterschiedlich langer Lerndauern auf die Effizienz der evolvierten Sensoren zu untersuchen. Lerndauer soll hier die Anzahl der Zeitschritte heißen, in denen jeder Agent sein individuelles Verhalten durch Q-Learning adaptieren kann. Die Anzahl der Erfolge während der Lerndauer bestimmt dann die reproduktive Fitness, die für Selektion in der Evolution verantwortlich ist. Um die Auswirkungen von unterschiedlichen Lerndauern sichtbar zu machen, muß ein neues von der Lerndauer unabhängiges Maß eingeführt werden, womit vergleichende Aussagen getroffen werden können: ungeachtet der vorangegangenen Lernphase wird das beste Individuum einer jeden Generation in einem eigenen Lauf mit konstanter Schrittzahl ausgewertet.

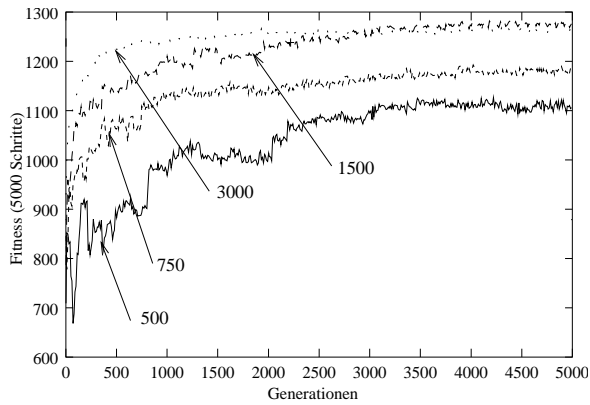
Das Ergebnis ist in Abbildung 7a)-c) für unterschiedliche Längen der Auswertungsphase angegeben. Für eine Länge von 5000 Zeitschritte (Abb. 7 a)) scheint die Vermutung bestätigt zu werden, daß mit wachsender Lerndauer die Erfolge in der konstanten Evaluierung anwachsen, also schneller erfolgreiche Sensorabbildungen gefunden werden. Dagegen beobachtet man für 1500 und 500 Schritte (Abb. 7 b)-c)) einen eher entgegengesetzten Trend: hier dominieren Individuen, die unter Bedingungen weniger Lernschritte evolviert worden sind.

Offensichtlich produzieren unterschiedliche Lernbedingungen verschiedene Typen von Sensoren, die eine deutlich niedrigere Fitness erzielen, falls das Testszenario stark vom Evolutionsszenario abweicht. Dies scheint die Vermutung zu bestätigen, daß der Lernprozess von zwei gegensätzlichen Faktoren beeinflusst wird: einerseits können Sensoren mit hoher Auflösung eher Zustände (Situationen) unterscheiden, in denen unterschiedliche Aktionen erforderlich sind und so eine höhere Fitness erzielen. Andererseits bedeutet ein kleine Menge möglicher interner Zustände eine kleinere Anzahl von Q -Werten, die aktualisiert werden müssen, so daß ein einzelner Q -Wert häufiger aktualisiert wird, was eine schnellere Adaption zur Folge hat.

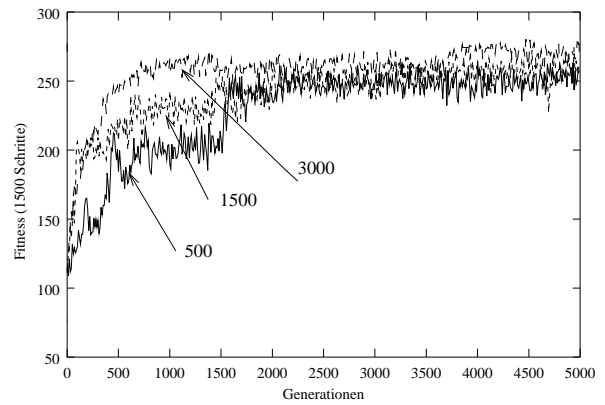
4.3 Quantitative Charakterisierung durch relevante Information

In einer zweiten Simulationsserie wurde versucht, die evolvierten Sensoren mittels der in Abschnitt 3 beschriebenen Entropiemaße zu charakterisieren. Die Plots der Ergebnisse in Abb. 7 d)-f) zeigen dabei den Evolutionsverlauf von relevanter und irrelevanter Information abhängig von den zugrundeliegenden Lernschritten.

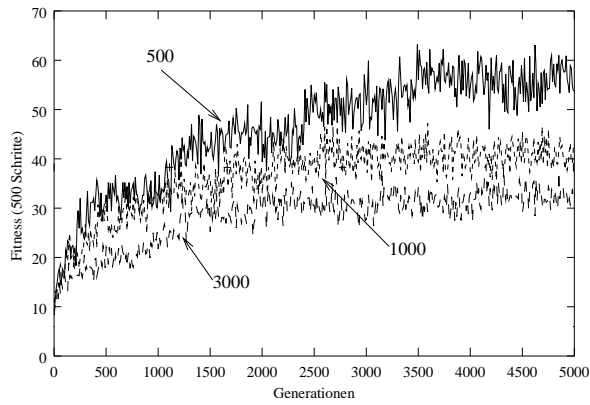
Abb. 7 d) und e) zeigen Verlauf von relevanter und irrelevanter Information. Offensichtlich neigen Individuen aus langen Lernphasen dazu, einen Großteil beider Arten von Information beibehalten, d.h. den Zustandsraum nur wenig zu komprimieren, wohingegen kurze Lernphasen einen bemerkenswerten Verlust sowohl relevanter als auch irrelevanter Information bedeutet.



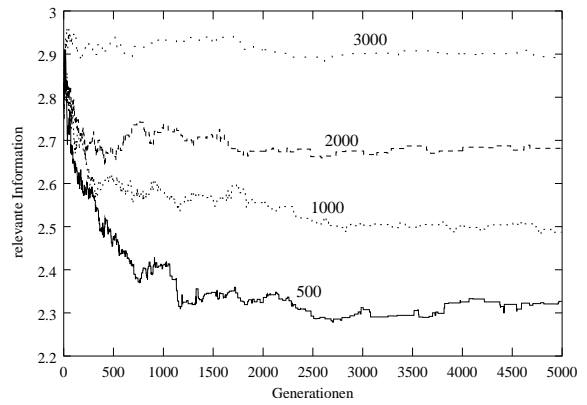
a) 5000 Evaluationschritte



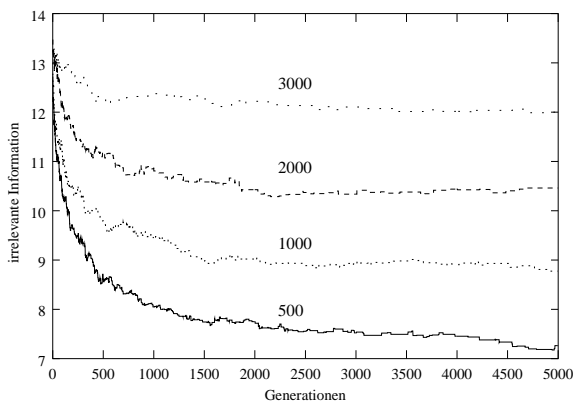
b) 1500 Evaluationschritte



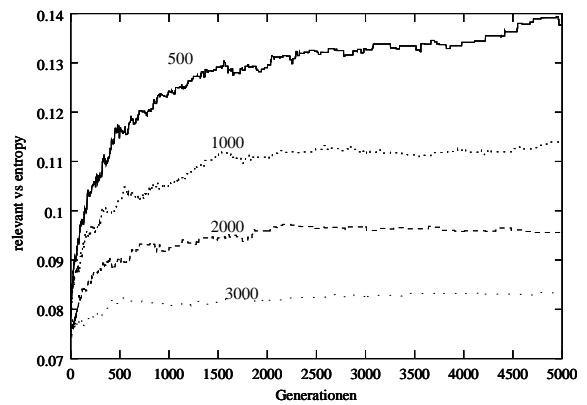
c) 500 Evaluationschritte



d) Relevante Information



e) Irrelevante Information



f) Verhältnis Relevant zu Gesamt

Abbildung 7: Fitness- und entropiebasierte Simulationsergebnisse, wobei Zahlen unter einzelnen Kurven die entsprechenden Lernschritte angeben. Beachte, daß Information nicht in "Bits" sondern in 2^{Bits} ausgedrückt ist. Alle angegebenen Werte wurden als Mittel aus 20 Läufen für das beste Individuum ermittelt.

Um den Anteil relevanter Information an der gesamten Entropie zu veranschaulichen, zeigt Abb. 7 f) das Verhältnis $2^{I(X:A)}/2^{H(X)}$. Hier wird deutlich, wie der Anteil relevanter Information für Sensoren kurzer Lernphasen stark ansteigt, während ihr Anteil bei lange lernenden Individuen nur schwach wächst.

Somit erfahren die qualitativen Beobachtungen eine neue Bedeutung, und die Schlußfolgerung liegt nahe, daß die Lerneffektivität von der Anzahl der internen Zustände in gegensätzlicher Weise abhängt: es gibt einen Trade-Off zwischen Komprimierung des Zustandsraumes, was die Lerngeschwindigkeit erhöht aber zum Informationsverlust neigt, und dem Beibehalten von Informationen, was vor allem bei Szenarien langer Lerndauer an Bedeutung gewinnt.

Literatur

- [1] K. Balakrishnan and V. Honovar. *On Sensor Evolution in Robotics*. In: Koza, Goldberg, Fogel, and Riolo (eds.) *Proceedings of 1996 Genetic Programming Conference – GP-96*; MIT Press, pp. 455–460. 1996.
- [2] J. M. Baldwin. *A new factor in evolution*. *American Naturalist*, **30** pp. 441-451, 1896.
- [3] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. *Acting Optimally in Partially Observable Stochastic Domains*. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.
- [4] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. *Planning and acting in partially observable stochastic domains*. *Artificial Intelligence*, **101**, 1998.
- [5] A. Liese, D. Polani, and T. Uthmann. *On the development of spectral properties of visual agent receptors through evolution*. In: D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, (eds.) *Proc. Genetic and Evolutionary Computation Conference (GECCO), Las Vegas, Nevada* pp. 857-864. Morgan Kaufmann, 2000.
- [6] T. Jung, P. Dauscher, T. Uthmann. *On Individual Learning, Evolution of Sensors and Relevant Information*. Accepted for Workshop "Evolution of sensors in nature, hardware, and simulation", GECCO 2001
- [7] A. Mark, D. Polani, and T. Uthmann. *A framework for sensor evolution in a population of braitenberg vehicle-like agents*. In: C. Adami, R. K. Belew, H. Kitano, and Ch. E. Taylor, (eds.) *Artificial Life VI*, pp. 428–432, MIT Press. 1998.
- [8] D. Polani, T. Martinetz, J. Kim. *On the Quantification of Relevant Information* Presented at the Seventh Scandinavian Conference on Artificial Intelligence, Odense, Denmark, February 19-21, 2001
- [9] C.E. Shannon. *A mathematical theory of communication*. *Bell System Technical Journal*, **27**:379–423,623–656, 1948.
- [10] S. Singh, T. Jaakkola, and M. I. Jordan. *Learning without state-estimation in partially observable markovian decision processes*. In: *Proceedings of the Eleventh Machine Learning Conference*, 1994.

- [11] R. Sutton, and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [12] N. Tishby, F. Pereira, and W. Bialek. *The Information Bottleneck method*. In *Proceedings of the 37th annual Allerton Conference on Communication, Control, and Computing 1999*.
- [13] M. Wall. *GAlib: A C++ Library of Genetic Algorithm Components*. <http://lancet.mit.edu/ga/>. (Januar 2001).