4.4 *Lernen und Aktionskoordinierung in Mehragentensystemen*

Gerhard Weiß

Dem Bereich des maschinellen Lernens in Mehragentensystemen wurde innerhalb der Verteilten Künstlichen Intelligenz lange Zeit keine oder nur sehr geringe Aufmerksamkeit geschenkt. Obwohl sich diese Situation inzwischen geändert hat und gerade in den letzten Jahren die Forschungsbemühungen deutlich verstärkt wurden, gibt es in diesem Bereich noch zahlreiche offene Fragen und Probleme. Dieser Beitrag untersucht den Zusammenhang zwischen Lernen und Aktionskoordinierung in Mehragentensystemen. Es wird ein Verfahren vorgestellt und untersucht, welches mehreren Agenten ermöglicht, ihre Aktionen zu koordinieren und geeignete Sequenzen von Aktionsmengen zu lernen.

4.4.1 Motivation

Das Konzept des *Mehragentensystems* ist in der Künstlichen Intelligenz und der Informatik keineswegs neu, sondern beeinflußte bereits vor vielen Jahren die anfänglichen Entwicklungen in Bereichen wie kognitive Modellierung [Sel59, Min79], Blackboard Systeme [EL75], objektorientierte Programmierung [Hew77] und Modellierung von Nebenläufigkeit [Pet62]. In den vergangenen Jahren lieferte dieses Konzept wichtige Impulse für neue Denkansätze in der Informatik (vgl. etwa [Bra91, BB92, LK91, Mal88]) und entwickelte sich zu einem Forschungsschwerpunkt in der Künstlichen Intelligenz (z.B. [BG88a, BH91, GH89, Huh87]). Das Interesse an solchen Systemen gründet vor allem auf der Tatsache, daß viele Aufgaben- und Problemstellungen – man denke etwa an autonome Fertigung oder Verkehrsflußüberwachung – von sich aus einen verteilten Lösungsansatz erfordern und besser mit mehreren anstatt mit einem einzelnen Agenten modelliert werden (vgl. [Geo87]). Insbesondere ermöglicht die Verwendung mehrerer Agenten sowohl eine bessere Handhabung von natürlichen Einschränkungen wie etwa die begrenzte Leistung eines einzelnen Agenten oder die physikalischen Verteiltheit der zu verarbeitenden Daten als auch die Nutzung von inhärenten Eigenschaften verteilter Systeme wie Parallelität, Robustheit und Fehlertoleranz.

Im Gegensatz zum Problemlösen und Planen wurde dem *verteilten Lernen* in Mehragentensystemen lange Zeit nur wenig Aufmerksamkeit geschenkt. Dies steht in deutlichem Widerspruch zu der mittlerweile gewonnen Einsicht, daß gerade dem Lernen eine wichtige Rolle zukommt: da nämlich solche Systeme typischerweise sehr komplex und deshalb in ihrem Verhalten nur schwer zu spezifizieren sind, sollten sie in der Lage sein, weitgehend selbständig zu lernen, die ihnen gestellten Aufgaben zu bewältigen. Erst seit wenigen Jahren gibt es verstärkt Forschungsbemühungen, die sich mit grundlegenden Aspekten des Mehragenten-Lernens beschäftigen. So wurden beispielsweise die Wechselwirkungen zwischen verteiltem und nichtverteiltem Lernen [Sia91a, SW89] näher untersucht und Verfahren zum verteilten Erlernen von Konzepten [BM91b] und von plausiblen Hypothesen [Sia90, Sia91b] entwickelt. Trotz dieser Bemühungen gibt es jedoch eine Vielzahl offener Fragen und Probleme im Bereich des Mehragenten-Lernens.

In diesem Beitrag wird der Zusammenhang zwischen Lernen und *Aktionskoordinierung* in Mehragentensystemen untersucht. Im Mittelpunkt stehen dabei folgende Fragestellungen:

- Wie können mehrere Agenten lernen, welche Aktionen nebenläufig ausgeführt werden sollen?
- Wie können mehrere Agenten lernen, welche Aktionsmengen sequentiell ausgeführt sollen?

Der Beitrag ist wie folgt strukturiert. Im Abschnitt 2 wird der Begriff eines Agenten konkretisiert und der Begriff einer Gruppe kompatibler Agenten eingeführt. Im Abschnitt 3 wird aufbauend auf diesen Agenten- und Gruppenbegriff ein Lernverfahren vorgestellt, welches auf die Beantwortung der beiden obigen Fragen abzielt. Im Abschnitt 4 werden theoretische und experimentelle Ergebnisse zu diesem Verfahren beschrieben. Im Abschnitt 5 werden mögliche Erweiterungen des vorgestellten Lernverfahrens aufgezeigt.

4.4.2 Agenten und Gruppen als Einheiten

In der Literatur findet sich eine Vielzahl unterschiedlicher Spezifikationen und Implementierungen von Mehragentensystemen. Diesem Artikel liegt die in der Verteilten Künstlichen Intelligenz „prototypischen Sichtweise" zugrunde, derzufolge ein Mehragentensystem aus mehreren autonomen Agenten besteht die in der Lage sind zu interagieren und die sich in ihren Fähigkeiten und ihrem Weltwissen voneinander unterscheiden. Dabei werden zwei Typen von organisatorischen Einheiten unterschieden: einzelne Agenten und Gruppen von Agenten die kompatible Aktionen ausführen.

Jeder *Agent* besitzt eine sensorische Komponente, eine motorische Komponente, eine Wissensbasis und eine Lerneinheit; siehe Bild 1. Insbesondere ist jeder Agent hinsichtlich seiner Aktivitäten in folgender Weise eingeschränkt:

- aufgrund beschränkter sensorischer Fähigkeiten kennt er nur einen bestimmten Ausschnitt seiner Umwelt,
- aufgrund beschränkter motorischer Fähigkeiten ist er nur zu bestimmten umweltbeeinflußenden Aktionen fähig und
- Aktionen verschiedener Agenten können inkompatibel sein.

Wegen dieser Einschränkungen ist es möglich, daß verschiedene Agenten unterschiedliches Wissen über ihre Umwelt besitzen und auf unterschiedliche und inkompatible Aktionen spezialisiert sind.

Die einzelnen Agenten dienen als Primitive („building blocks") für komplexere Organisations- und Kontrollstrukturen. Eine der elementarsten Strukturen ist eine *Gruppe* von Agenten die kompatible Aktionen ausführen [Fox81, Gal73]. Diese elementare Struktur liegt auch dem im nächsten Abschnitt beschriebenen Lernverfahren zugrunde. Der Gruppenbegriff ist wie folgt definiert. Eine Gruppe besteht aus einem Gruppenleiter und mehreren kompatiblen Gruppenmitgliedern, wobei ein Leiter ein einzelner Agent und jedes Mitglied entweder ein

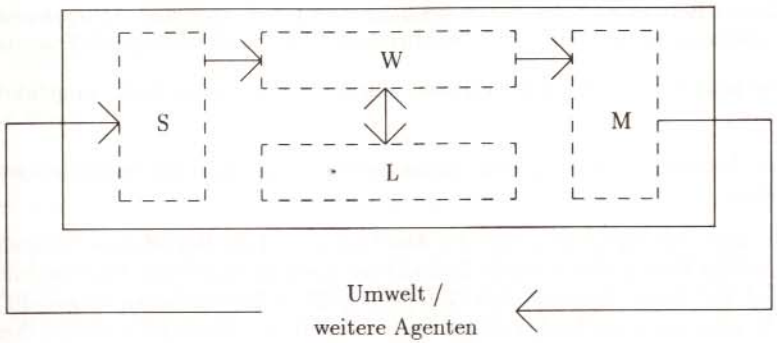

Bild 1: Struktur eines Agenten. Jeder Agent besteht aus einer sensorischen Komponente (S), einer motorischen Komponente (M), einer Wissensbasis (W) und einer Lerneinheit (L). Die sensorische und motorische Komponente erlauben dem Agenten die Interaktion mit seiner Umwelt. Die Wissensbasis enthält das Wissen des Agenten über seine Umwelt. Die Lerneinheit modifiziert die Wissensbasis derart, daß der Agent zu einem verbesserten Verhalten befähigt wird.

einzelner Agent oder wiederum eine Gruppe ist. Dabei bedeutet „kompatibel", daß in einem gegebenen Umweltzustand die Aktivität keines der Mitglieder zu Umweltveränderungen führt, welche die Aktivität eines anderen Mitgliedes verhindern. Der Gruppenleiter hat die Aufgabe, die Interessen seiner Gruppe zu vertreten; insbesondere bewertet er die Zielrelevanz seiner Gruppe und entscheidet, ob seine Gruppe weiterhin als autonome Einheit agiert, Mitglied einer anderen Gruppe wird oder aufzulösen ist (siehe Abschnitt 3). Die Anzahl der von einer Gruppe ausführbaren umweltmodifizierenden Aktionen wird nachfolgend das Aktionspotential einer Gruppe genannt. Dieses Aktionspotential stellt zusammen mit dem Hierarchisierungsgrad die charakteristischen Merkmale einer Gruppe dar. Wie Bild 2 illustriert, erlaubt diese rekursive Definition die Bildung von einfachen als auch von komplex strukturierten Gruppen.

In den nächsten Abschnitten wird folgende Sprechweise und Notation verwendet. Der Aktivitätskontext eines Agenten in einem gegebenen Umweltzustand ist definiert als derjenige Ausschnitt des Umweltzustandes, welcher dem Agenten bekannt ist. In jedem Umweltzustand ist der Aktivitätskontext einer Gruppe definiert als die Summe der Aktivitätskontexte aller in dieser Gruppe enthaltenen Agenten. Ein Agent ist potentiell aktiv in einem Umweltzustand, wenn er seine Aktion ausführen könnte; eine Gruppe ist potentiell aktiv, wenn alle beteiligten Agenten potentiell aktiv sind. Ein Agent und eine Gruppe ist autonom in einem Umweltzustand, wenn er bzw. sie nicht Mitglied einer in diesem Umweltzustand potentiell aktiven Gruppe ist. S_j bezeichnet einen Umweltzustand. U_i bezeichnet eine organisatorische Einheit, also einen Agenten oder eine Gruppe. Falls U_i eine Gruppe ist, dann verweist $\overline{U_i}$ auf den Leiter dieser Gruppe; falls U_i ein einzelner Agent ist, dann bezeichnet $\overline{U_i}$ ebenfalls diesen Agenten. Falls U_i eine Gruppe ist und $U_{i_1}, \ldots, U_{i_n}$ ihre Mitglieder sind, so wird dies kurz mit $U_i = \langle U_{i_1}, \ldots, U_{i_n} \rangle$ ausgedrückt. Schließlich bezeichnet $[U_i, S_j]$ den Aktivitätskontext der

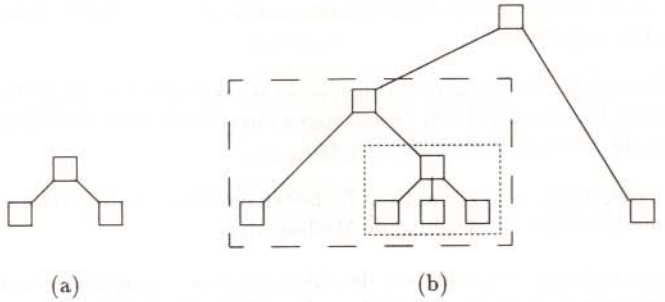

Bild 2: Struktur einer Gruppe. Eine Gruppe besteht aus einer Menge von einzelnen Agenten (□), die durch Leiter–Mitglied Relationen (/ \) hierarchisch strukturiert sind. *(a)* zeigt eine Gruppe, die nur aus dem Leiter und zwei Mitgliedern besteht, wobei jedes Mitglied ein einzelner Agent ist. *(b)* zeigt eine Gruppe, die einen Agenten und eine weitere Gruppe (gestricheltes Kästchen) als Mitglieder hat; die „gestrichelte Gruppe" ihrerseits hat einen Agenten und eine Gruppe (gepunktetes Kästchen) als Mitglieder, wobei die „gepunktete Gruppe" ihrerseits drei einzelne Agenten als Mitglieder hat. Alle Mitglieder einer Gruppe müssen kompatibel sein.

Einheit U_i im Zustand S_j.

4.4.3 Der DFG Algorithmus

Prinzipielle Arbeitsweise

Der DFG Algorithmus – DFG als Akronym für „Dissolution and Formation of Groups" – ist ein Verfahren zum Erlernen von geeigneten Sequenzen von Aktionsmengen in Mehragentensysstemen [Wei92b]. Er basiert auf der aktionsorientierten Variante [Wei92a] des aus dem Bereich der sog. classifier systems stammenden bucket brigade Lernprinzips [Hol86] und stellt eine Weiterentwicklung der in [Wei93] beschriebenen Lernverfahren zur verteilten Aktionskoordinierung dar. Der Algorithmus gehört zur Klasse der Reinforcement–Lernverfahren, das heißt, die von der Umwelt erforderliche Lernrückkoppelung ist minimal und besteht lediglich aus zu gewissen Zeitpunkten bereitgestellten skalaren Werten.

Die prinzipielle Arbeitsweise des DFG Algorithmus kann wie folgt zusammengefaßt werden. Es wird zwischen einzelnen Agenten und Gruppen bestehend aus mehreren kompatiblen Agenten als den agierenden organisatorischen Einheiten unterschieden (siehe Abschnitt 2). Jede Einheit bewertet die Relevanz ihrer Aktivität in Abhängigkeit vom jeweiligen Umweltzustand. Lernen erfolgt durch zwei sich wechselseitig beeinflußende Prozesse: Bewertungsmodifizierung (credit assignment), also die Modifizierung der Aktivitätsbewertungen, und Gruppenentwicklung, also der Bildung neuer und der Zerfall alter Gruppen. In jedem Umweltzustand findet abhängig von den Aktivitätsbewertungen ein Wettbewerb zwischen den Einheiten statt. Nur dem Gewinner eines Wettbewerbes ist es gestattet, tatsächlich aktiv zu werden und damit den aktuellen in den nächsten Umweltzustand zu überführen. Insgesamt

ergibt sich die Gesamtaktivität des Mehragentensystems durch die wiederholte Ausführung des folgenden Arbeitszyklus:

1. Aktivitätsauswahl: Die Einheiten prüfen, ob sie im aktuellen Umweltzustand aktiv werden könnten. Basierend auf den Bewertungen ihrer Aktivitäten wird diejenige Einheit bestimmt, die tatsächlich aktiv werden darf.

2. Bewertungsmodifizierung: Die aktiven Einheiten modifizieren ihre Aktivitätsbewertungen entsprechend dem bucket brigade Mechanismus.

3. Gruppenentwicklung: Abhängig von den Aktivitätsbewertungen werden alte (erfolglose) Gruppen aufgelöst und neue (erfolgreichere) Gruppen gebildet.

Im folgenden werden die einzelnen Schritte dieses Zyklus im Detail beschrieben.

Aktivitätsauswahl und Bewertungsmodifizierung

In jedem Umweltzustand S_j läuft zwischen den Agenten ein Wettbewerb um das Recht, aktiv zu werden. Für jede potentiell aktive und autonome Einheit U_i berechnet $\overline{U_i}$ ein Gebot B_i^j gemäß

$$B_i^j = (\alpha + \beta) \cdot E_i^j \quad , \tag{4.1}$$

wobei α eine Konstante, β ein Zufallsterm und E_i^j ein skalarer Wert ist, der die von $\overline{U_i}$ vorgenommene Bewertung der Aktivität von U_i im Kontext $[U_i, S_j]$ darstellt. (Die Autonomiebedingung gewährleistet, daß die Mitglieder einer potentiell aktiven Gruppe nicht miteinander konkurrieren. Der Zufallsterm dient der Vermeidung lokaler Lernminima; es sei hier erwähnt, daß im Bereich der classifier systems verschiedene Varianten des zufallsbeeinflußten Wettbewerbs vorgeschlagen wurden.) Die Aktivitätsauswahl erfolgt dann über die Bestimmung der höchstbietenden Einheit. Nur diese Einheit, also der Gewinner dieses Wettbewerbs, wird aktiv und generiert somit einen neuen Umweltzustand. (Die Überprüfung auf potentielle Aktivität und Autonomie erfordert nur Kommunikation entsprechend der Leiter–Mitglied Relationen; beispielsweise wird eine Gruppe $U_i = \langle U_{i_1}, \ldots, U_{i_n} \rangle$ von $\overline{U_i}$ als „potentiell aktiv" deklariert, falls jedes Mitglied U_{i_j} von $\overline{U_{i_j}}$ als „potentiell aktiv" deklariert wurde.)

Die Bewertungsmodifizierung erfolgt durch eine lokale Umverteilung der Bewertungen zwischen den aktiven Einheiten. Sei U_i der Gewinner im aktuellen Zustand S_j und sei U_k der Gewinner im unmittelbar vorausgehenden Zustand S_l. $\overline{U_i}$ reduziert seine Bewertung E_i^j um den deterministischen Anteil $\alpha \cdot E_i^j$ seines Gebots und zahlt diesen Anteil an $\overline{U_k}$; dieser erhöht dann seine eigene Bewertung E_k^l um den erhaltenen Betrag. (Der aktuelle Gewinner zahlt für das Previleg, aktiv zu werden, der vorherige Gewinner wird dafür belohnt, die Umwelt in geeigneter Weise modifiziert zu haben.) Gibt es zudem eine Lernrückkoppelung R^{ext} von der Umwelt, so addiert diese der aktuelle Gewinner zur seiner eigenen Bewertung. Insgesamt ergeben sich also folgende Modifikationen:

$$E_i^j = E_i^j - \alpha \cdot E_i^j + R^{ext} \tag{4.2}$$

$$E_k^l = E_k^l + \alpha \cdot E_i^j \quad . \tag{4.3}$$

Informell lassen sich sich Auswirkungen dieser Modifikationen wie folgt beschreiben (vgl. auch [Hol85]). In einer Sequenz von aktiven Einheiten zahlt jede Einheit einen gewissen Betrag an ihren Vorgänger und erhält einen gewissen Betrag von ihrem Nachfolger. Zum einen nimmt dadurch die Bewertung einer Aktivität einer Einheit zu (ab), wenn eine Einheit weniger (mehr) bezahlt als sie erhält. Zum anderen wird jede Bewertungsänderung im Laufe der Zeit innerhalb einer Sequenz „nach hinten weitergereicht". Dies führt zu einer Stabilisierung einer Sequenz, falls die letzte Einheit regelmäßig eine positive Lernrückkoppelung von der Umwelt erhält, und zu einer Destabilisierung, falls dies nicht der Fall ist.

Gruppenentwicklung

Gruppenentwicklung umfaßt zwei gegenläufige Prozesse: den Zerfall alter und die Formation neuer Gruppen. Beide Prozesse gehorchen folgenden Entwicklungsprinzipien:

- Anfangs ist kein Agent und keine neu gebildete Gruppe bereit, mit anderen Einheiten zu kooperieren und neue Gruppen zu bilden (d.h. Gruppenmitglied zu werden).
- Eine Einheit ist nicht bereit zu kooperieren, solange die Bewertung ihrer Aktivität im Zuge der Bewertungsmodifizierung eine ansteigende Tendenz aufweist.
- Eine Einheit ist bereit zu kooperieren, sobald die Bewertung ihrer Aktivität nur mehr geringfügig ansteigt, stagniert oder sogar abfällt.
- Kompatible und kooperationswillige Einheiten bilden neue Gruppen.
- Eine Gruppe wird aufgelöst und zerfällt damit in ihre Mitglieder, falls die Bewertung ihrer Aktivität unter ein kritisches Minimum abfällt.

Diese Prinzipien sind folgendermaßen realisiert. Um die zeitliche Entwicklung und die Tendenz der Bewertung einer Einheit U_i beurteilen zu können, berechnet $\overline{U_i}$ den gleitenden Mittelwert seiner Bewertungen über die vergangenen Episoden, wobei eine Episode definiert ist als das Zeitintervall zwischen zwei aufeinanderfolgenden externen Lernrückkoppelungen. Formal bedeutet dies, daß $\overline{U_i}$ während jeder Episode $\tau + 1$ den Mittelwert $M_i^j[\tau + 1]$ seiner Bewertung E_i^j berechnet gemäß

$$M_i^j[\tau + 1] = \frac{1}{\nu} \cdot \sum_{T=\tau-\nu+1}^{\tau} E_i^j[T] \quad , \tag{4.4}$$

wobei ν eine Konstante ist, welche den zu berücksichtigenden vergangenen Zeitabschnitt bestimmt, und $E_i^j[T]$ die Bewertung E_i^j am Ende der Episode T ist. Mit Hilfe dieses Mittelwertes lassen sich nun Kriterien für den Zerfall und die Bildung von Gruppen spezifizieren.

Sei $\tau+1$ die aktuelle Episode, S_j der aktuelle Umweltzustand und U_i eine in S_j potentiell aktive und autonome Einheit. Dann entscheidet $\overline{U_i}$, daß U_i im Kontext $[U_i, S_j]$ bereit ist zu kooperieren und mit anderen Einheiten eine neue Gruppe zu bilden, falls

$$M_i^j[\tau + 1] \leq \sigma \cdot E_i^j[\tau - \nu] \quad , \tag{4.5}$$

wobei σ eine Konstante ist, welche das Maß der Kooperationsbereitschaft steuert. Unter allen kooperationsbereiten Einheiten wählt zunächst diejenige Einheit, die mit der höchsten Bewertung assoziiert ist, ihre Kooperationspartner. Hierbei sind verschiedene Auswahlstrategien denkbar. In den gegenwärtigen Implementierung wird nur ein einzelner Partner (also ein Agent oder eine Gruppe) gewählt, und zwar derjenige, dessen Aktivität kompatibel ist und am höchsten bewertet wird. Diese Auswahl wird iteriert, bis keine weiteren Gruppenneubildungen möglich sind.

Umgekehrt entscheidet $\overline{U_i}$, daß seine Gruppe U_i aufzulösen ist, falls

$$M_i^j[\tau+1] \leq \rho \cdot E^{init} \quad , \tag{4.6}$$

wobei ρ eine Konstante ist, welche die Auflösungsbereitschaft der Gruppen reguliert. (Die Autonomiebedingung spielt dabei eine zweifache Rolle: im Falle der Gruppenbildung verhindert sie die Kooperationsbereitschaft einer Einheit, falls sie bereits Mitglied einer potentiell aktiven Gruppe ist, und im Falle des Gruppenzerfalls verhindert sie die Auflösung einer Gruppe, falls sie Mitglied einer potentiell aktiven Gruppe ist.)

Die Bildung und der Zerfall von Gruppen stellen äußerst kontextsensitive Prozesse dar. Dies ist auch wünschenswert, da die Aktivität einer Gruppe in unterschiedlichen Umweltzuständen typischerweise auch unterschiedliche Bewertungen erfordert.

4.4.4 Analyse

Theoretische Überlegungen

Da der DFG Algorithmus auf das Erlernen geeigneter Sequenzen von Aktionsmengen abzielt, ist entscheidend, wie die Bewertungen von Aktivitäten aufeinanderfolgend aktiver Einheiten verändert werden. Erweitert man das in [Gre88] beschriebene Konvergenzergebnis für den bucket brigade Algorithmus auf den DFG Algorithmus, so erhält man folgendes

Resultat 1 (Lernkonvergenz). Sei $\{U_{i_1}, \ldots, U_{i_n}\}$ eine Menge von organisatorischen Einheiten derart, daß

(i) U_{i_k}, $1 \leq k < n$, mit $U_{i_{k+1}}$ „gekoppelt" ist, d.h. immer wenn U_{i_k} in einem Kontext $[U_{i_k}, S_{j_k}]$ aktiv ist, dann ist im darauffolgenden Zyklus $U_{i_{k+1}}$ im Kontext $[U_{i_{k+1}}, S_{j_{k+1}}]$ aktiv, und

(ii) nur U_{i_n} eine eventuelle Lernrückkoppelung von der Umwelt erhält.

Dann gilt: falls $E_{i_n}^{j_n}$ gegen einen konstanten Wert E^* konvergiert, dann konvergiert auch $E_{i_k}^{j_k}$, $1 \leq k < n$, gegen diesen Wert.

Beweis. Sei U_{i_k}, $1 \leq k < n$, während des Zyklus t aktiv. Dann folgt:

$$E_{i_k}^{j_k}[t+2] = E_{i_k}^{j_k}[t] - \alpha \cdot E_{i_k}^{j_k}[t] + \alpha \cdot E_{i_{k+1}}^{j_{k+1}}[t+1] \quad ,$$

wobei $E_{i_k}^{j_k}[t+2]$ den Wert von $E_{i_k}^{j_k}$ zu Beginn des Zyklus $t+2$ bezeichnet. Daraus läßt sich die Gleichung

$$E_{i_k}^{j_k}[\overline{s}+2] = (1-\alpha)^n \cdot E^{init} + + \textstyle\sum_{r=1}^{s} \alpha \cdot (1-\alpha)^{s-r} \cdot E_{i_{k+1}}^{j_{k+1}}[\overline{r}+1] \quad ,$$

ableiten, wobei $s \in \mathsf{N}$ und $\overline{s}$ definiert ist als $\overline{s} = t$ genau dann wenn U_{i_k} während des Zyklus

t zum s-ten Mal im Kontext $[U_{i_k}, S_{j_k}]$ aktiv ist. Konvergiert nun $E_{i_{k+1}}^{j_{k+1}}$ gegen E^*, dann ergibt sich

$$\lim_{t\to\infty} E_{i_k}^{j_k}[t] = lim_{s\to\infty} E_{i_k}^{j_k}[\overline{s}+2] = E^* \; . \qquad \blacksquare$$

Unter dem DFG Algorithmus konvergieren also die Bewertungen aufeinanderfolgend aktiver Einheiten gegen ein Gleichgewichtsniveau. Unter Gleichgewichtsbedingungen zahlt dann jede Einheit an ihren Vorgänger denselben Betrag, den sie von ihrem Nachfolger erhält.

Entscheidend ist auch die Frage nach der Güte der gelernten Lösungen, wobei eine Lösung wie üblich definiert ist als jede Sequenz $S_1, \ldots, S_n$ von Umweltzuständen mit S_1 = Startzustand, S_n = Zielzustand und $S_i \neq S_n$ für alle $i \in \{1, \ldots, n-1\}$. Eine Antwort auf diese Frage liefert folgendes

Resultat 2 (Lösungsqualität). Jede unter dem DFG Algorithmus gelernte Lösung ist zyklenfrei.

Beweis. Sei $S_1, \ldots, S_n$ eine gelernte Lösung die nicht zyklenfrei ist. Dann gibt es ein $i \in \{1, \ldots, n-2\}$ und ein $k \in \{1, \ldots, n-i\}$ mit $S_i = S_{i+k}$. Somit ist in S_i und S_{i+k} dieselbe organisatorische Einheit aktiv. Daraus folgt unmittelbar $S_{i+1} = S_{i+k+1}$, oder allgemeiner, $S_{i+m} = S_{i+k+m}$ für jedes $m \in \{1, \ldots, n-i-k\}$. Somit ist $S_{n-k} = S_n$, und dies ist im Widerspruch zur Annahme, daß $S_1, \ldots, S_n$ eine Lösung darstellt. $\blacksquare$

Dieses Ergebnis zeigt insbesondere, daß die „globale Eigenschaft" der Zyklenfreiheit erreicht wird, obwohl jeder einzelne Agent nur einen lokalen Ausschnitt des jeweiligen Umweltzustandes kennt und kein expliziter Wissenstransfer zwischen den Agenten erfolgt.

Experimentelle Ergebnisse

Um erste Erfahrungen mit dem DFG Algorithmus zu sammeln, wurde die blocksworld als Experimentierumgebung gewählt. Im folgenden werden die Ergebnisse zu der in Bild 3 gezeigten Aufgabe beschrieben. Jeder Agent ist in eine bestimmte Tätigkeit spezialisiert (z.B. kann Agent A_1 den Block A auf den Boden stellen und der Agent A_2 kann Block A auf Block B legen). Vorbedingung für die Ausführbarkeit einer Aktion $put(x, y)$ ist, daß keine anderen Blöcke auf x und y positioniert sind. Jeder Agent besitzt in jedem Umweltzustand nur minimales Wissen: er weiß nur, ob die Vorbedingung seiner Aktion erfüllt ist. Aufgrund dieser Einschränkung ist ein Agent nicht in der Lage, alle verschiedenen Umweltzustände zu differenzieren; vielmehr kann er nur zwischen der Klasse der Zustände, in denen er seine Aktion ausführen könnte, und der Klasse der Zustände, in denen dies nicht der Fall ist, unterscheiden (siehe auch [Wei93]). Diese Einschränkung erschwert die Aufgabe erheblich (und differenziert sie maßgeblich von der „klassischen" Sichtweise dieses Aufgabentyps), da nun kein Agent zu irgendeinem Zeitpunkt einen „globalen Überblick" über die Blockkonstellation besitzt. Insbesondere ist es nun möglich, daß ein Agent nicht mehr in der Lage ist, zwischen Zuständen, in denen seine Aktion nützlich ist, und Zuständen, in denen sie nutzlos ist, zu unterscheiden. (Es sei hier explizit darauf hingewiesen, daß die Beschränkung eines Agenten auf eine einzelne Aktion vom DFG Algorithmus nicht gefordert wird; vielmehr dient sie hier dem Zweck, das notwendige Umweltwissen jedes einzelnen Agenten in plausibler Weise minimal zu halten.) Je zwei Agenten gelten als kompatibel, wenn ihre Aktionen nicht folgende Bedingungen verletzen: in keinem Umweltzustand

Startkonfiguration Zielkonfiguration

Agenten: A_1: $put(A, \perp)$ A_2: $put(A, B)$ A_3: $put(A, F)$ A_4: $put(B, C)$
A_5: $put(C, D)$ A_6: $put(D, E)$ A_7: $put(E, \perp)$ A_8: $put(E, F)$
A_9: $put(F, \perp)$ A_10: $put(G, H)$ A_11: $put(H, \perp)$

Zeitintervall: maximal 5 Zyklen

Bild 3: Eine vorgegebene Menge von Agenten soll lernen, innerhalb eines begrenzten Zeitintervalls die Start- in die Zielkonfiguration zu transformieren.

- dürfen verschiedene Blöcke auf dieselbe Position gestellt werden
- darf ein Block auf verschiedene Positionen gestellt werden
- darf ein Block einen anderen Block gestellt werden, dessen Position in diesem Zustand bereits verändert wurde.

Es wird vorausgesetzt, daß den einzelnen Agenten diese Kompatibilitätsbedingungen bekannt sind. (Die erste Bedingung verhindert Konflikte aufgrund sich unmittelbar ausschließender Aktionen. Die beiden anderen Bedingungen halten zum einen die zwischen den Agenten erforderliche Kommunikation gering und ermöglichen zum anderen eine gezielte Bewertung von einzelnen Aktionen.) Beispiele für inkompatible Agenten sind A_3 und A_8, A_2 und A_3, und A_5 und A_6. Mehrere Agenten gelten kompatibel, wenn sie paarweise kompatibel sind.

Der Schwierigkeitsgrad der in Bild 3 gezeigten Aufgabe wird durch eine Analyse des Suchraums deutlich. Aus den 11 Einzelaktionen lassen sich unter Berücksichtigung der Kompatibilitätsbedingungen 40 verschiedene Gruppen mit Aktionspotential 2, 89 verschiedene Gruppen mit Aktionspotential 3 und 119 verschiedene Gruppen mit Aktionspotential 4 bilden. Es gibt nur eine einzige Lösung der Länge 2, 62 Lösungen der Länge 3, 1168 Lösungen der Länge 4 und 11940 Lösungen der Länge 5 (wobei eine Lösung definiert ist als eine Sequenz von Mengen kompatibler Aktionen, die die Start- in die Zielkonfiguration überführt). Keine Lösung enthält weniger als 6 Einzelaktionen, weshalb diese Aufgabe ohne Gruppenbildung nicht lösbar ist. Die Wahrscheinlichkeit, daß eine zufällig gewählte und anwendbare Sequenz der Länge 2 (3, 4, 5) die Aufgabe löst, beträgt 0.1 (0.8, 1.4, 1.6) Prozent; insgesamt ist damit die Wahrscheinlichkeit, daß eine zufällig gewählte und anwendbare Sequenz der Maximallänge 5 die Aufgabe löst, geringer als 4 Prozent.

Bild 4 zeigt die Lernresultate des DFG Algorithmus für folgende Parameterkonstellation: $\alpha = 0.15$, $\beta \in [-\alpha/5 \ldots + \alpha/5]$ (zufällig generiert), $\nu = 4$, $\sigma = 1 + 3\alpha$, $\rho = 1 - \alpha$, und

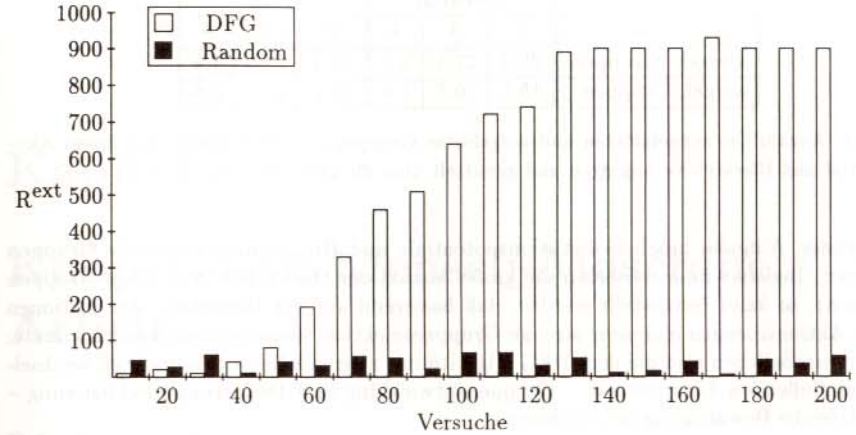

Bild 4: Lernperformanz des DFG Algorithmus.

$E^{init} = R^{ext} = 1000$. (Weitere experimentelle Resultate finden sich z.B. in [Wei92b].) Jeder Datenwert zeigt, gemittelt über 10 Läufe mit unterschiedlicher Initialisierung des Pseudozufallszahlengenerators, die in den jeweils letzten 10 Versuchen erzielte durchschnittliche externe Lernrückkoppelung. Ein Versuch ist dabei definiert als jede Sequenz bestehend aus maximal 5 Arbeitszyklen, welche zur Lösung der Aufgabe führt (erfolgreicher Versuch), und jede Sequenz der exakten Länge 5, welche die Aufgabe nicht löst (erfolgloser Versuch). Nach jedem Versuch wird die Startkonstellation erneut präsentiert. Die externe Lernrückkoppelung wird nur am Ende eines erfolgreichen Versuches bereitgestellt. Wie die Lernkurve zeigt, liegt die erzielte Lernperformanz deutlich über der zufälligen Performanz. Dies illustriert insbesondere, daß der DFG Algorithmus mehreren Agenten ermöglicht, geeignete und stabile Lösungen zu lernen, obwohl jeder einzelne Agent nur ein sehr beschränktes Wissen von seiner Umwelt besitzt.

Zum Verständnis dieser Lernresultate ist es notwendig, die vom DFG Algorithmus erzeugten Gruppen näher zu betrachten. Gemittelt über die 10 Läufe wurden durchschnittlich pro Lauf 59.9 neue Gruppen gebildet und 27.2 existierende Gruppen aufgelöst. Wie in Abschnitt 2 erwähnt wurde, stellen das Aktionspotential und der Hierarchisierungsgrad die entscheidenden Merkmale einer Gruppe dar. In Tabelle 1 ist dargestellt, wieviele der neugebildeten und aufgelösten Gruppen das Aktionspotential $x \in \{2, 3, 4\}$ und den Hierarchisierungsgrad $y \in \{2, 3, 4\}$ besitzen, und zwar ebenfalls gemittelt über die 10 Läufe. (Entsprechend der Aufgabe ist x und somit y stets kleiner oder gleich 4.) Dabei liegt folgende übliche Definition des Hierarchisierungsgrades zugrunde:

$$\mathrm{Grad}(U_i) = \begin{cases} 1 & \text{falls } U_i \text{ ein einzelner Agent ist} \\ \max_j\{\mathrm{Grad}(U_{i_j})\} + 1 & \text{falls } U_i = \langle U_{i_1}, \ldots, U_{i_n} \rangle \end{cases}$$

Wie diese Tabelle zeigt, gelingt es dem DFG Algorithmus, zu allen im Rahmen der zu

	Potential			Grad		
	2	3	4	2	3	4
neugeb. Gruppen	28.4	27.3	4.2	28.4	29.6	1.9
aufgel. Gruppen	15.1	10.5	1.6	15.1	11.5	0.6

Tabelle 1: Anzahl der neugebildeten und aufgelösten Gruppen, aufgeschlüsselt nach ihrem Aktionspotential und Hierarchisierungsgrad und gemittelt über die 10 Läufe.

bewältigenden Aufgabe möglichen Aktionspotentiale und Hierarchisierungsgrade Gruppen zu erzeugen. Bedenkt man weiterhin die große Anzahl der theoretisch möglichen Gruppen (siehe oben), so kann festgestellt werden, daß basierend auf der Bewertung der Aktionen bzw. der Aktionsmengen eine sehr strenge Gruppenselektion vorgenommen werden konnte. Insgesamt ermöglichen also die dem DFG Algorithmus zugrundeliegenden und sich wechselseitig beeinflußenden Lernprozesse – Gruppenentwicklung und Bewertungsmodifizierung – eine erfolgreiche Bewältigung des Suchraums.

4.4.5 Schlußbetrachtungen

Mit dem DFG Algorithmus wurde ein allgemeines Verfahren zum Erlernen von geeigneten Sequenzen von Aktionsmengen in Mehragentensystemen vorgestellt. Dieses Verfahren liefert eine breite Grundlage für weiterführende Untersuchungen im Umfeld des verteilten Lernens und der Aktionskoordinierung.

Ausgehend von den bisher gewonnenen Ergebnissen lassen sich folgende zentrale Themen hinsichtlich möglicher Erweiterungen des DFG Algorithmus ableiten:

- Der Lernerfolg des DFG Algorithmus ist an eine explizite Bewertung genügend vieler Umweltzustände Aktionsmengen gebunden. Welche in der Künstlichen Intelligenz entwickelten Methoden zur Generalierung, Wissensakquistion und Planung können dazu verwendet werden, diese Bindung zu mindern?

- Das dem DFG Algorithmus zugrundeliegende Gruppenkonzept impliziert eine strenge Hierarchie zwischen den Agenten. Welche alternativen und möglicherweise flexibleren Gruppenkonzepte sind anwendbar?

- Die Gruppenentwicklung erfolgt statisch nach fest vorgegebenen Kriterien. Welche alternativen und möglicherweise adaptiven Kriterien und Strategien für den Zerfall und die Bildung von Gruppen sind anwendbar?

Diese Fragestellungen müssen Gegenstand zukünftiger Forschung sein. Da bei ihrer Beantwortung weitgehend Neuland in der Künstlichen Intelligenz und in der Informatik betreten wird, ist es zweckmäßig und ratsam, auch einschlägige Literatur aus anderen Wissenschaftsbereichen wie etwa der Sozialpsychologie (z.B. [Guz82, Lau88]) und den Wirtschaftswissenschaften (z.B. [AS78, Gal73, HLM85, SS90], die sich mit solchen oder verwandten Fragestellungen schon seit vielen Jahren beschäftigen, heranzuziehen.