Dezentrale Marktmechanismen zur Förderung kooperativen Verhaltens in selbstinteressierten Multiagentensystemen

Kyrill Schmid

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Kyrill Schmid

München, den 11.08.2022

1. Berichterstatter/in:	Prof. Dr. Claudia Linnhoff-Popien
2. Berichterstatter/in:	Prof. Dr. Stefan Fischer
Tag der Einreichung:	11.08.2022
Tag der Disputation:	30.03.2023

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, 11.08.2022 Kyrill Schmid

Danksagung

Im Folgenden möchte ich mich bei einigen Personen bedanken, die mich auf unterschiedliche Art und Weise gefördert, motiviert und unterstützt haben und somit wesentlich dazu beigetragen haben, dass ich diese Arbeit erstellen konnte.

Zuerst möchte ich Prof. Dr. Claudia Linnhoff-Popien danken! Liebe Claudia, die Jahre am Lehrstuhl Mobile und Verteilte Systeme waren für mich eine spannende und äußerst lehrreiche Phase! Vielen Dank, dass du mir die Gelegenheit gegeben hast unter deiner Betreuung zu promovieren - es war eine tolle Zeit für mich!

Des Weiteren gilt mein herzlicher Dank Herrn Prof. Dr. Stefan Fischer, der sich freundlicherweise bereit erklärt hat, die Rolle des Zweitberichterstatters zu übernehmen!

Ganz herzlich möchte ich mich auch bei Prof. Dr. Lenz Belzner bedanken! Lieber Lenz, dass du mich von Beginn an inhaltlich begleitet und motiviert hast, war eine sehr hilfreiche und inspirierende Erfahrung!

Darüber hinaus möchte ich all meinen Kollegen und Kolleginnen vom Lehrstuhl Mobile und Verteilte Systeme danken, die stets zur Verfügung gestanden haben, wenn es darum ging inhaltliche Fragen zu allen möglichen Themen ausgiebig zu diskutieren und die in guten wie auch schwierigeren Zeiten für mich da waren!

Vielen Dank!

Zusammenfassung

Der Forschungszweig kooperative künstliche Intelligenz hat in den letzten Jahren zunehmend Aufmerksamkeit erfahren. Das liegt nicht zuletzt daran, dass KI-basierte Systeme mehr und mehr Einzug in unseren Alltag nehmen und sich somit die Frage stellt, wie sich nicht nur intelligente Verhaltensweisen erzeugen lassen, sondern auch sozialverträgliche Systeme entwickeln lassen. Die Fähigkeit zu kooperativem Handeln ist dabei von zentraler Bedeutung, da fehlende Kooperationsbereitschaft dramatische Auswirkungen haben kann. Dies gilt insbesondere für Anwendungsfälle in denen gemeinsame Ressourcen genutzt werden, wobei fehlendes kooperatives Verhalten zu ineffizienter Ressourcennutzung oder gar zur Übernutzung von Ressourcen führen kann, so dass eine nachhaltige Nutzung nicht gewährleistet ist. Eine Möglichkeit selbstlernende KI-Einheiten zu kooperativem Handeln zu incentivieren, liegt in dem Einsatz von Marktmechanismen. Dabei wird die Eigennützigkeit der Agenten genutzt, um basierend auf bilateralem Austausch Pareto-Verbesserungen zwischen den Agenten zu realisieren. Da jeder Akteur freiwilliger Teilnehmer eines Marktes sein kann, ist jeder Handel eine Verbesserung, von der mindestens zwei Seiten profitieren. Die realisierten Transaktionen zwischen den Agenten sorgen dafür, dass die geteilten Ressourcen effizienter allokiert, d.h. der Menge der Agenten zugewiesen werden. In dieser Arbeit werden unterschiedliche Marktmechanismen betrachtet, die sich in der Art und Weise unterscheiden, unter welchen Bedingungen Transaktionen abgeschlossen werden können. Zur Modellierung eines lernenden Systems werden Methoden des Reinforcement Learning eingesetzt, wobei jeder Agent als eigenständige Instanz eines Lernalgorithmus repräsentiert wird. Die Evaluationen erfolgen in unterschiedlichen Zwei- und Mehrspieler-Umgebungen, die sich auch in dem potentiellen Konfliktpotential zwischen den Agenten unterscheiden.

Abstract

The research field of cooperative artificial intelligence has received increasing attention in recent years. This is not least due to the fact that AI-based systems are finding their way more and more into our everyday lives, raising the question of how not only intelligent behaviour can be generated, but also how socially acceptable systems can be developed. The ability to act cooperatively is of central importance here, as a lack of willingness to cooperate can have dramatic consequences. This is especially true for use cases in which shared resources are used, whereby a lack of cooperative behaviour can lead to inefficient resource use or even to the overuse of resources, so that sustainable use is not guaranteed. One way to incentivise self-learning AI units to act cooperatively is to use market mechanisms. Here, the self-interest of the agents is used to realise Pareto improvements between the agents based on bilateral exchange. Since every agent can be a voluntary participant in a market, every trade is an improvement from which at least two sides benefit. The realised transactions between agents ensure that the shared resources are allocated more efficiently, i.e. allocated to the set of agents. In this work, different market mechanisms are considered, which differ in the way under which conditions transactions can be concluded. Reinforcement learning methods are used to model a learning system, where each agent is represented as an independent instance of a learning algorithm. The evaluations take place in different two- and multi-player environments, which also differ in the potential for conflict between the agents.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Zugrundeliegende Vorarbeiten 2
1.2 Aufbau dieser Arbeit 5

2 Definitionen und Grundlagen **7**
2.1 Spieltheoretische Grundlagen 7
2.1.1 Normalform-Spiele 8
2.1.2 Nash-Gleichgewicht 9
2.1.3 Soziale Dilemmas 10
2.1.4 Sequentielle Soziale Dilemmas 11
2.1.5 Verhandlungsspiele 12
2.2 Reinforcement Learning 14
2.2.1 Markov-Entscheidungsprozess 15
2.2.2 Werte-Funktionen 16
2.2.3 Bellman-Gleichung 16
2.2.4 Wertebasierte Methoden 17
2.2.5 Approximative Methoden 19
2.2.6 Gradientenbasierte Methoden 24
2.2.7 Verteilungsbasiertes Reinforcement Learning 24
2.2.8 Herausforderungen des Reinforcement Learning 25
2.3 Multiagent Reinforcement Learning 26
2.3.1 Multiagentensysteme 27
2.3.2 Stochastisches Spiel 29
2.3.3 Unabhängiges Lernen 29
2.3.4 Metriken 30

3 Marktmechanismen in stochastischen Spielen **33**
3.1 Vorveröffentlichungen 33
3.2 Stochastische Marktspiele 34
3.2.1 Verwandte Arbeiten 35
3.2.2 Konzept 37
3.2.3 Bedingungslose Märkte 38
3.2.4 Konditionale Märkte 39
3.2.5 Training unabhängiger Agenten mit Markt 40
3.2.6 Einsatzmöglichkeiten eines Marktes 42
3.2.7 Zusammenfassung und Diskussion 45
3.3 Verhandlungsprozesse 47

3.3.1 Verwandte Arbeiten 48
3.3.2 Konzeption zweier Verhandlungs-Domänen 49
3.3.3 Modellierung risikoaverser Agenten 53
3.3.4 Experimente 57
3.3.5 Zusammenfassung und Diskussion 61

4 Konditionale und bedingungslose Marktmechanismen 63
4.1 Vorveröffentlichungen 63
4.2 Konditionaler Markt: Action Trading 63
4.2.1 Idee und Motivation 64
4.2.2 Verwandte Arbeiten 65
4.2.3 Konzept 66
4.2.4 Action Trading für Normalformspiele mit zwei Spielern 68
4.2.5 Action Trading in sequentiellen sozialen Dilemmas 71
4.3 Bedingungsloser Markt: Shareholding 80
4.3.1 Idee und Motivation 80
4.3.2 Verwandte Arbeiten 81
4.3.3 Konzept 82
4.4 Vergleich Action Trading und Shareholding 85
4.4.1 Evaluationsumgebung 86
4.4.2 Agenten und Training 88
4.4.3 Ergebnisse 89
4.4.4 Diskussion 91
4.5 Zusammenfassung 92

5 Vertragsbasierte Marktmechanismen und Kooperation in sozialen Dilemmas 93
5.1 Vorveröffentlichungen 93
5.2 Vertragsbasierte Marktmechanismen 94
5.2.1 Idee und Motivation 94
5.2.2 Verwandte Arbeiten 95
5.2.3 Konzept 97
5.2.4 Evaluation 103
5.3 Kooperation in Sozialen Dilemmas 114
5.3.1 Idee und Motivation 114
5.3.2 Verwandte Arbeiten 116
5.3.3 Ansatz 116
5.3.4 Evaluation 118
5.4 Zusammenfassung 127

6 Zusammenfassung und Ausblick 129

Abkürzungsverzeichnis 133

Abbildungsverzeichnis 136

1 Einleitung

Kooperation zwischen eigennützig agierenden Individuen ist eine Herausforderung im Kontext zwischenmenschlicher Interaktion, erlangt aber auch zunehmend Relevanz im Bereich autonomer Systeme [21]. Die Verbreitung und der Einsatz von künstlicher Intelligenz in unserem täglichen Leben bringt die Frage mit sich, inwiefern diese Systeme Verhaltensweisen hervorbringen, die aus einer normativen Sichtweise wünschenswert sind. Denkt man beispielsweise an autonom fahrende Autos im Straßenverkehr, so ist es höchstwahrscheinlich nicht ausreichend, wenn autonome Einheiten lediglich die geltenden Gesetze bei der Entscheidungsfindung berücksichtigen. Vielmehr sollten sie auch in der Lage sein, situativ kooperativ zu handeln, da sich durch kooperatives Verhalten auf lokaler Ebene oftmals größere Probleme reduzieren oder gar verhindern lassen (wie beispielsweise Staubildung). Kooperation muss nicht zwangsläufig aus rein uneigennützigen Motiven resultieren. So ist es ebenfalls möglich, dass sich durch Kooperation alle beteiligten Individuen gleichzeitig besser stellen im Vergleich zu einer Situation ohne Kooperation. Wenn die Möglichkeit besteht, dass die Akteure ihre durch die Kooperation entstehenden Gewinne untereinander frei austauschen können, so kann es bereits ausreichen, wenn lediglich ein einzelner Teilnehmer von einer Kooperation profitiert, da er die anderen Kooperateure an dem entstehenden Gewinn durch Transaktionen beteiligen kann.

Solange jedoch alle Entscheidungsträger individuelle Ziel-Metriken optimieren, die die Auswirkungen des eigenen Handelns auf den Erfolg anderer Agenten nicht berücksichtigen, sind die Ziele der anderen Agenten *exogen* und werden daher bei der eigenen Entscheidungsfindung nicht berücksichtigt. Unter diesen Umständen ist die Entstehung kooperativer Verhaltensweisen unwahrscheinlich, insbesondere dann, wenn zusätzlich Wettbewerb, beispielsweise durch das Teilen von Ressourcen zwischen den Agenten besteht [45]. Eine Möglichkeit zur *Internalisierung* der Ziele anderer Akteure für das Individuum liegt in der Integration von Marktmechanismen. Ein Markt erlaubt es selbstinteressierten und autonom lernenden Agenten, mittels Austausch von Ressourcen untereinander, die Ziele der Agenten aneinander anzugleichen und somit die negativen Folgen der individuellen Optimierung aufzulösen.

Märkte stellen sowohl aus theoretischer als auch aus praktischer Sicht ein effektives Mittel dar, um eine große Anzahl von eigennützigen Einheiten zu koordinieren und können unter bestimmten Bedingungen dazu führen, dass sich ein Pareto-optimales Ergebnis einstellt. Neben Attributen wie Offenheit und Skalierbarkeit sind Märkte besonders effektiv bei der Förderung effizien-

ter, kooperativer Interaktion zwischen heterogenen Teilnehmern, da (idealisierte) Märkte besondere Regeln (Eigentumsrechte, fälschungssichere Währungen, Handelsmarken) definieren, so dass kooperatives Verhalten zu einer dominanten Strategie wird. Märkte wurden darüber hinaus schon früh zur Koordination verteilt arbeitender Prozesse für rechenintensive Jobs vorgeschlagen [55].

Während Märkte im KI-Kontext bereits verwendet wurden, um verteilte Entscheidungsfindung in vollständig kooperativen Multiagentensystemen zu koordinieren [18, 33, 55], werden in dieser Arbeit Marktmechanismen zwischen selbstinteressierten Individuen untersucht, die das Erlernen kooperativen Verhaltens ermöglichen. Durch die Integration eines Marktes kann die individuelle Nutzenmaximierung von einem potentiellen systemischem Fehler zu einem Vorteil werden, da das marktbasierte Handeln immer gleichzeitig alle beteiligten Akteure besser stellt. Die Anreize individueller Agenten werden dadurch derart verändert, so dass Kooperation zu einer vorteilhaften Strategie wird und defektives also unkooperatives Verhalten an Attraktivität verliert. Durch die Ermöglichung eines freien Austausches von Ressourcen wird jeder Agent des Systems zum freiwilligen Marktteilnehmer, wodurch sein selbst-interessiertes Handeln zu einer wünschenswerten Eigenschaft wird, welche dazu beiträgt, dass der Austausch effizient funktioniert.

Motiviert durch den erfolgreichen Einsatz von Reinforcement Learning in Multiagentensystemen [47, 64, 87] werden in dieser Arbeit unterschiedliche Ansätze des Reinforcement Learning zum Training der Agenten verwendet. Durch die empirische Evaluation der marktbasierten Ansätze in unterschiedlichen Evaluationsumgebungen stellt sich heraus, dass sich die Integration von Peer-to-Peer basiertem Handel zwischen den Agenten sowohl positiv auf das erzielte Gesamtergebnis als auch das individuell erreichte Ergebnis auswirkt.

1.1 Zugrundeliegende Vorarbeiten

Die in dieser Arbeit zusammengefassten Forschungsinhalte wurden bereits zu überwiegendem Teil auf entsprechenden Fachkonferenzen publiziert und vor internationalem Fachpublikum präsentiert. Um eine genaue Zuordnung von bereits publizierten Ergebnissen zu den entsprechenden Inhaltskapiteln zu ermöglichen, werden im Folgenden entsprechende Details von für die Arbeit relevante Publikationen gegeben und auch auf den Eigenanteil des Autors dieser Arbeit (im Folgenden: der Autor) eingegangen. Die Aufzählung erfolgt dabei in der Reihenfolge der Referenzierung in den nachfolgenden Inhaltskapiteln, wann immer dies möglich ist. Folgende Personen waren an den einzelnen Veröffentlichungen beteiligt: Prof. Dr. Claudia Linnhoff-Popien (Ludwig-Maximilians-Universität München), Prof. Dr. Lenz Belzner (Technische Hochschule Ingolstadt), Dr. Thomas Gabor (Ludwig-Maximilians-Universität München), Thomy Phan (Ludwig-Maximilians-Universität München), Robert Müller (Ludwig-Maximilians-Universität München), Johannes Tochtermann (Ludwig-Maximilians-Universität München).

- **Action Markets in deep Multiagent Reinforcement Learning [74]** In dieser Arbeit wurde der erste marktbasierte Ansatz vorgestellt. Bei diesem Ansatz, genannt Action Trading, können Agenten durch Erweiterungen ihrer Aktionsräume Transaktionen miteinander durchführen. Das Handeln ist konditional, eine Transaktion wird also nur durchgeführt, wenn Leistung und Gegenleistung erfolgen. Die Idee des Ansatzes stammt vom Autor, der auch die Implementierung und Evaluation durchgeführt hat. Lenz Belzner hat die Arbeit von Beginn an begleitet, stand stets für Diskussionen zur Verfügung und hat wertvolles Feedback gegeben. Zu den Eigenanteilen des Autors zählen neben der eigentlichen Idee auch die Ausarbeitung der Veröffentlichung. Thomas Gabor und Thomy Phan waren ebenfalls in verschiedene Diskussionen involviert und unterstützten bei der Ausarbeitung der Publikation durch Feedback. Inhalte dieser Publikation finden sich in Kapitel 3, in dem die Idee des Ansatzes erläutert wird und in Kapitel 4, in dem der Ansatz explizit beschrieben und evaluiert wird.

- **Multiagent Reinforcement Learning for Bargaining under Risk and Asymmetric Information [77]** Kern dieser Arbeit ist die Konzeption sowie Evaluation von zwei verschiedenen Verhandlungssituationen zwischen selbstinteressierten und unabhängig trainierten Agenten. Dabei werden verschiedene Einflüsse wie Risikoaversion und Informationsasymmetrie auf das Verhandlungsergebnis untersucht, wobei die Agenten mittels deep Reinforcement Learning trainiert werden. Die Idee des Ansatzes stammt vom Autor, der auch die Implementierung und Evaluation durchgeführt hat. Die gemeinsamen Diskussionen und das Feedback zur Ausarbeitung des Konzepts von Lenz Belzner, Thomy Phan und Thomas Gabor waren dabei sehr hilfreich. Claudia Linnhoff-Popien war ebenfalls an der konzeptionellen Entwicklung der Veröffentlichung beteiligt. Inhalte dieser Publikation finden sich in Kapitel 3, genauer in Abschnitt 3.3, wobei die entwickelten Verhandlungsdomänen und die Evaluation dazu beschrieben werden.

- **Stochastic Market Games [76]** In dieser Veröffentlichung wird das Konzept des stochastischen Marktspiels als formale Erweiterung des stochastischen Spiels vorgestellt. Dabei werden auch die beiden Marktformen (konditional und bedingungslos) eingeführt und anhand zweier konkreter Instanzen erläutert und evaluiert. Die Idee des stochastischen Marktspiels als Erweiterung des stochastischen Spiels entstand während verschiedener Diskussionen mit Lenz Belzner. Die Idee der konditionalen und bedingungslosen Marktmechanismen stammt vom Autor, der auch die Implementierung und Evaluation durchgeführt hat. Bei der Ausarbeitung der Veröffentlichung waren außerdem Robert Müller, Johannes Tochtermann sowie Claudia Linnhoff-Popien beteiligt, wodurch wertvolles Feedback und Verbesserungsvorschläge zustande kamen. Inhalte die-

ser Publikation finden sich in Kapitel 3, in dem die Definition des stochastischen Marktspiels gegeben wird, sowie in Kapitel 4, in dem der bedingungslose Marktmechanismus des *Shareholding* erläutert und evaluiert wird.

- **Distributed Emergent Agreements with Deep Reinforcement Learning [78]** In dieser Arbeit wird die Idee zur Möglichkeit von Verträgen zwischen den Agenten beschrieben und evaluiert. Verträge realisieren dabei bilaterale Abkommen zwischen den Agenten und regeln die Aktionswahl für die Laufzeit des Vertrages sowie die daraus resultierenden Auszahlungen für die beteiligten Agenten. Die Idee dazu stammt vom Autor, der auch die Implementierung und Evaluation durchgeführt hat. Neben zahlreichen Gesprächen und Diskussionen waren Robert Müller, Lenz Belzner, Johannes Tochtermann auch in die Ausarbeitung der Veröffentlichung involviert, deren wertvolles Feedback wesentlich zum Erfolg der Arbeit beigetragen hat. Auch Claudia Linnhoff-Popien war an der konzeptionellen Entwicklung der Veröffentlichung beteiligt. Inhalte dieser Veröffentlichung finden sich in Kapitel 5, in dem der Ansatz *Distributed Emergent Agreements with Deep Reinforcement Learning* ausführlich beschrieben und evaluiert wird.

- **Learning to Penalize Other Learning Agents [75]** In dieser Arbeit wird die Anwendbarkeit der marktbasierten Mechanismen auf soziale Dilemmas untersucht. Der Ansatz wurde dabei so abgewandelt, dass dadurch anstelle einer positiven Incentivierung eine Bestrafung der Agenten untereinander möglich ist. Die Idee dazu stammt vom Autor, der auch die Implementierung und Evaluation durchgeführt hat. Lenz Belzner war von Beginn an in zahlreiche Diskussionen involviert, wodurch der Erfolg der Arbeit substantiell beeinflusst wurde. An der Ausarbeitung der Veröffentlichung war auch Claudia Linnhoff-Popien beteiligt. Die Inhalte dieser Publikation sind in zweiten Abschnitt des Kapitels 5 beschrieben.

Zur besseren Übersicht werden im Folgenden die in den Inhaltskapiteln thematisierten Kerninhalte den entsprechenden Publikationen zugeordnet:

Die Kerninhalte des ersten Abschnitts von Kapitel 3, das die Definition des stochastischen Marktspiels zum Inhalt hat, sowie die Konzeption und Evaluation von Verhandlungssituationen, wurden bereits in [76] bzw. in [77] veröffentlicht. Die Abgrenzung von konditionalen und bedingungslosen Märkten (siehe Abschnitt 3.2) findet sich in [76]. Ebenso die Beschreibung des Trainingsverfahrens unabhängiger Agenten mit Marktmechanismus (siehe Abschnitt 3.2.5), sowie die Auslotung der potentiellen Einsatzmöglichkeiten eines Marktes (siehe Abschnitt 3.2.6) wurden in [76] beschrieben. Der zweite Abschnitt von Kapitel 3 behandelt Verhandlungssituationen zwischen unabhängigen Agenten (siehe Abschnitt 3.3). Die Inhalte wurden bereits in [77] veröffentlicht.

In Kapitel 4 wird jeweils ein konditionaler und ein bedingungsloser Marktmechanismus beschrieben, wobei der konditionale Ansatz (siehe Abschnitt 4.2)

in [74] veröffentlicht wurde und der bedingungslose Ansatz (siehe Abschnitt 4.3) in [76] veröffentlicht wurde. Der Vergleich der beiden Verfahren (siehe Abschnitt 4.4) wurde ebenfalls in [76] veröffentlicht.

Kapitel 5 gliedert sich in zwei Abschnitte, wobei der erste Abschnitt vertragsbasierte Marktmechanismen behandelt (siehe Abschnitt 5.2). Der Ansatz *Distributed Emergent Agreement Learning (DEAL)* (siehe Abschnitt 5.2.3) wurde dabei bereits in [78] veröffentlicht. Die Evaluation dazu mit zwei Agenten (siehe Abschnitt 5.2.4) ist noch nicht vorveröffentlicht worden. Die Evaluation mit mehr als zwei Agenten (siehe Abschnitt 5.2.4) wurde ebenfalls in [78] veröffentlicht. Der zweite Abschnitt von Kapitel 5 befasst sich mit Kooperation in sozialen Dilemmas (siehe Abschnitt 5.3). Sämtliche Inhalte darin wurden bereits in [75] vorveröffentlicht.

1.2 Aufbau dieser Arbeit

Diese Arbeit ist wie folgt aufgebaut: In Kapitel 2 werden relevante Grundlagen aus den Bereichen der Spieltheorie, des Reinforcement Learning und des Multiagent Reinforcement Learning eingeführt, die für das weitere Verständnis der Arbeit relevant sind. In Kapitel 3 erfolgt eine allgemeine Beschreibung des Konzepts, in das die später beschriebenen Ansätze eingeordnet werden können. In Kapitel 3 werden darüber hinaus die Einsatzmöglichkeiten eines Marktmechanismus in einem Multiagentensystem ausgelotet. Darüber hinaus erfolgt eine Beschreibung einer weiteren relevanten Klasse von Situationen zwischen selbstinteressierten Agenten, die als Verhandlungsspiele bezeichnet werden.

Kapitel 4 beschreibt zwei konkrete Marktmechanismen, die als konditional und bedingungslos bezeichnet werden. Nach der jeweiligen Motivation zu dem Ansatz erfolgt eine Beschreibung der spezifischen Marktfunktionen sowie eine Evaluation der Ansätze. In Kapitel 5 werden vertragsbasierte Marktmechanismen vorgestellt und evaluiert, sowie eine Erweiterung des konditionalen Marktansatzes auf soziale Dilemmas beschrieben.

Den Abschluss bildet Kapitel 6 mit einer Zusammenfassung und einem Ausblick für die behandelten Themen.

2 Definitionen und Grundlagen

Diese Arbeit bewegt sich an der Schnittstelle verschiedener Disziplinen. Zentrale Frage der Arbeit ist, wie sich in Systemen, die sich aus unabhängigen und autonom lernenden Einheiten zusammensetzen, ein höherer Grad an Kooperation zwischen den Akteuren erwirken lässt. Da die Frage von Kooperation zwischen eigennützig handelnden Akteuren traditionell in der Spieltheorie behandelt wird, werden relevante Modelle und Konzepte eingeführt. Die Spieltheorie liefert sowohl eine Menge an Modellen wie beispielsweise soziale Dilemmas, die sich zur Evaluation von Ansätzen eignen, als auch verschiedene Lösungskonzepte, wie das Nash-Gleichgewicht, das als Referenzmodell verwendet werden kann. Der Lernvorgang einzelner Individuen (auch Agenten genannt), wird in dieser Arbeit durch Techniken des Reinforcement Learning umgesetzt. Reinforcement Learning bietet ein Rahmenwerk, das neben einer Vielzahl bekannter Lernalgorithmen auch eine profunde mathematische Basis für die Interaktion eines Agenten mit einer unbekannten Lernumgebung liefert. Darüber hinaus werden Definitionen und Konzepte aus dem Bereich der Multiagentensysteme und des Multiagent Reinforcement Learning benötigt, um Systeme parallel lernender Agenten zu untersuchen. Die nächsten Abschnitte geben einen Überblick zu den spieltheoretischen Grundlagen, den verwendeten Konzepten und Techniken des Reinforcement Learning und Multiagent Reinforcement Learning, sowie zu Multiagentensystemen.

2.1 Spieltheoretische Grundlagen

Die Spieltheorie umfasst Methoden zur Analyse von Interaktionen unabhängiger und eigennützig motivierter Entscheidungsträger [61]. Dabei wird generell von rationalen Akteuren ausgegangen, d.h. dass die Entscheidungsträger wohldefinierten exogenen Zielsetzungen folgen und sich darüber hinaus strategisch verhalten, also den Effekt ihrer Handlungen auf andere Akteure mit in die eigene Entscheidung einfließen lassen. Laut dieser Definition sind KI-basierte Agenten, so wie sie in dieser Arbeit verwendet werden, nur als bedingt rational im Sinne der Spieltheorie zu betrachten: Jeder Agent hat zwar ein klar definiertes Ziel, das in der Maximierung seiner individuellen Belohnungen liegt, jedoch werden keine strategischen Modelle in Bezug auf andere Akteure verwendet oder explizit erlernt. Die in dieser Arbeit verwendeten Modelle der Agenten lernen erst während des Trainingsprozesses, welche Aktionen in einer gegebenen Situation optimal sind und wie andere Agenten ihr Verhalten als Reaktion darauf verändern. Die begrenzte Rationalität resultiert also aus der

fehlenden Möglichkeit, sich über die strategischen Auswirkungen ihres Handelns im Vorhinein klar zu sein und dem Unvermögen dieses Wissen somit strategisch einsetzen zu können.

Spieltheoretische Modelle sind oftmals hochgradig abstrahierte Modelle, bei denen die Akteure über einen stark limitierten Aktionsraum verfügen und die Interaktion auf einer einzigen singulären Begegnung mit einem anderen Entscheidungsträger beruht. Diese Vereinfachung erlaubt den Einsatz rigoroser mathematischer Methoden zur Analyse von Gleichgewichten und dem Ableiten von Handlungsvorschriften. Es lassen sich viele für die Praxis relevante Situationen in Form abstrakter Modelle darstellen, wie sie in Wirtschaft, Politik, zwischenmenschlichen Beziehungen oder in Gesellschaftsspielen auftreten. Zentrale Frage der Spieltheorie ist die Identifikation von Gleichgewichten d.h. Zuständen, die stabil sind, so dass kein Akteur einen Anreiz hat, sein Verhalten ausgehend von diesem Zustand zu ändern. Eines der wichtigsten Konzepte zur Modellierung von Interaktionen zwischen mehreren Agenten ist das Normalform-Spiel, das im Folgenden Abschnitt beschrieben wird.

2.1.1 Normalform-Spiele

Das Normalform-Spiel (auch strategisches Spiel genannt) stellt die Interaktion einer endlichen Menge von Spielern $\mathcal{N}$ dar, bei der jeder Spieler i simultan eine Aktion aus einer endlichen Menge von Aktionen, dem sogenannten Aktionsraum $\mathcal{A}^i$ wählt. Durch die Aktionswahl aller Spieler ergibt sich daraus die zusammengesetzte Aktion $\mathbf{a} = (a_j)_{j\in\mathcal{N}}$, die auch als Profil bezeichnet wird (engl. *Joint Action* genannt). Die Menge aller Profile A ergibt sich aus dem kartesischen Produkt aller individuellen Aktionsräume $A = \times_{j\in\mathcal{N}}\mathcal{A}^j$.

Um rationale Entscheidungen eines Spielers ableiten zu können, muss jeder Spieler i eine Präferenz-Relation $\succeq_i$ besitzen, die über der Menge der Profile A definiert ist, da die Ergebnisse eines Spiels nicht nur durch die eigene Aktionswahl beeinflusst werden, sondern ebenfalls durch die Aktionswahl aller anderen Spieler. In vielen Fällen kann die Präferenz-Relation für Spieler i in Form einer Nutzenfunktion repräsentiert werden $u_i : A \to \mathbb{R}$, so dass gilt $u_i(a) \geq u_i(b)$ wenn $a \succeq_i b$, d.h. Spieler i präferiert Ergebnis a gegenüber b, wenn der Nutzen aus a größer ist als aus b. Insgesamt umfasst ein Normalform-Spiel also die Menge der Spieler $\mathcal{N}$, deren Aktionsräume $(\mathcal{A}^i)_{i\in\mathcal{N}}$ und deren Nutzenfunktionen $(u_i)_{i\in\mathcal{N}}$.

Im Fall von zwei Spielern können Normalform-Spiele kompakt in Form einer Matrix-Darstellung repräsentiert werden (siehe Grafik 2.1 für ein Zwei-Spieler-Spiel). Die Aktionen des ersten Spielers werden dabei in den Zeilen, die des zweiten Spielers in den Spalten dargestellt. Die durch ein Komma getrennten Werte in Zelle (i, j) geben die Belohnungen der Spieler an, die aus der Wahl eines Aktionstupels resultieren, wobei die erste Komponente die Belohnung von Spieler 1 beschreibt und die zweite Komponente die Belohnung von Spieler 2. In Grafik 2.1 führt beispielsweise die Wahl der Aktionen (a_1, a_1) zu einer

	a_1	a_2
a_1	w_1, w_2	x_1, x_2
a_2	y_1, y_2	z_1, z_2

Abbildung 2.1: Zwei Spieler Normalform-Spiel mit zwei Aktionen in Matrixform.

Belohnung von w_1 für Spieler 1 und w_2 für Spieler 2.

2.1.1.1 Iterative Spiele in Normalform

Wird ein Normalform-Spiel nicht nach einmaliger Aktionswahl der Spieler beendet, sondern für eine beliebige Anzahl an Iterationen wiederholt, so spricht man von einem iterativen Spiel in Normalform (engl. *Iterated Game*). Das Wissen darüber, dass ein Spiel mit dem gleichen Gegenüber mehrfach wiederholt wird, kann die Entscheidungen der Akteure maßgeblich beeinflussen. So kann es sein, dass aufgrund der zeitlich verlängerten Interaktion von eigennützigen Spielern kooperatives Verhalten entsteht [7]. Diese Form der Zusammenarbeit kann durch sogenannte Reziprozität entstehen, also durch das Wissen der Spieler, dass der Gegenspieler in zukünftigen Wiederholungen des Spiels unter Umständen kooperiert bzw. Vergeltungsschläge durchführt, in Abhängigkeit davon, ob man selbst zum aktuellen Zeitpunkt kooperativ oder unkooperativ handelt.

2.1.2 Nash-Gleichgewicht

Das gebräuchlichste Lösungskonzept der Spieltheorie ist das Nash-Gleichgewicht. Ein Nash-Gleichgewicht beschreibt einen stabilen Zustand eines Normalform-Spiels von dem aus keiner der Spieler mehr einen Anreiz hat seine Strategie zu ändern. Formal lässt sich ein Nash-Gleichgewicht als ein Aktionstupel $a^* \in A$ definieren, so dass für jeden Spieler $i \in \mathcal{N}$ gilt:

$$(a^*_{-i}, a^*_i) \succeq_i (a^*_{-i}, a_i) \forall a_i \in A$$

Dabei bezeichnet a_i die Aktion von Spieler i und a_{-i} die Aktionen aller anderen Spieler ohne i. Es gilt also, das a^* genau dann ein Nash-Gleichgewicht ist, wenn kein Spieler i eine Aktion wählen kann, so dass ein Ergebnis erzielt wird, das er gegenüber dem Ergebnis unter a^* präferiert. Es kann sich also kein Spieler durch einseitiges Abweichen von a^* besserstellen, solange die anderen Spieler ihre Strategien beibehalten. Generell gilt, dass nicht jedes Normalform-Spiel ein Nash-Gleichgewicht haben muss, es können aber prinzipiell auch mehrere

Nash-Gleichgewichte existieren.

2.1.3 Soziale Dilemmas

Eine wichtige Klasse von Problemen und elementarer Forschungsgegenstand spieltheoretischer Analysen sind soziale Dilemmas. Soziale Dilemmas beschreiben Situationen, bei denen es einen Bruch gibt zwischen individuell rationalem Verhalten und kollektiv rationalem Verhalten. Ein bekanntes Beispiel einer solcher Situationen ist die Tragik der Allmende. Dabei geht es um eine Situation, in der mehrere Individuen eine gemeinsame Ressource teilen wie beispielsweise Weideflächen, Fischbestände in öffentlichen Gewässern oder andere natürliche Ressourcen wie Holz. In diesen Situationen ist es aus Sicht des Individuums unter Umständen stets rational, weitere Einheiten der Ressource zu verbrauchen, da die Kosten nicht individuell sondern gemeinschaftlich getragen werden, wodurch die gemeinsame Ressource allerdings übermäßig beansprucht wird. Das Dilemma entsteht dadurch, dass es aus kollektiver Sicht optimal wäre, wenn die Ressource nur in solchem Maße genutzt wird, dass die Grenzkosten einer weiteren verbrauchten Einheit nicht den Grenznutzen, der dem Individuum aus dem Gebrauch der Ressource entsteht, übertreffen. Da die Kosten der Nutzung aber von allen getragen werden, der Nutzen aber individuell bezogen wird, gilt die Gleichung von Grenzkosten und Grenznutzen nicht, was zur Übernutzung führt. Handelt es sich um lebendige Ressourcen wie beispielsweise Fischbestände, die nur nachwachsen, wenn noch genügend Fische leben, so kann durch die Übernutzung eine völlige Vernichtung der Ressource entstehen. Kollektiv wäre es daher wünschenswert, wenn die Ressource nur bis zu einem gewissen Grad genutzt wird und somit eine kontinuierliche Erneuerung der Rohstoffe möglich ist. Folgen die Individuen lediglich eigennützigen Strategien, ist eine nachhaltige Nutzung oftmals nur schwierig zu realisieren.

2.1.3.0.1 Das Prisoner's Dilemma Andere soziale Dilemmas modellieren die Interaktion zwischen zwei Individuen, die Wahl haben miteinander zu kooperieren (gekennzeichnet durch die Aktion C) oder nicht zu kooperieren (gekennzeichnet durch die Aktion D). Das wohl bekannteste soziale Dilemma der Spieltheorie ist das *Prisoner's Dilemma*, dargestellt in Grafik 2.2.

	C	D
C	$1, 1$	$-0.5, 1.5$
D	$1.5, -0.5$	$0, 0$

Abbildung 2.2: Das Prisoner's Dilemma

Im Prisoner's Dilemma können die Entscheidungsträger entweder gemeinsam kooperieren, wodurch beide Agenten eine Belohnung von 1 erhalten. Durch einseitiges Abweichen von der kooperativen Aktion, kann sich der unkooperative Agent jedoch einen Vorteil gegenüber der Situation von beidseitiger Kooperation verschaffen. Somit ist Gier, also die individuelle Bereicherung auf Kosten des Gegenübers ein Motiv für nicht-kooperatives Verhalten im Prisoner's Dilemma. Da beide Agenten davon ausgehen müssen, dass der andere Agent sich für unkooperatives Handeln entscheidet, wenn man selbst kooperiert, würde sich ein rationaler Spieler von vornherein dazu entschließen, selbst unkooperativ zu handeln, um somit seinen potentiellen Schaden zu minimieren. Somit wird auch die Angst davor hintergangen zu werden, zu einem Motiv für unkooperatives Verhalten in diesem Spiel. Das einzige Nash-Gleichgewicht des Prisoner's Dilemma liegt daher darin, dass beide Agenten nicht kooperieren, obwohl sie sich insgesamt beide durch Kooperation besserstellen könnten.

2.1.3.0.2 Soziale Dilemmas mit mehr als zwei Spielern Soziale Dilemmas können auch mehr als zwei Akteure umfassen. Diese lassen sich weiter unterteilen in Spiele öffentlicher Güter (engl. *Public Good Games*) und Allmendegut-Spiele. Erstere beschreiben Situationen, in denen die Akteure gemeinsam investieren müssen, damit ein öffentliches Gut zur Verfügung gestellt werden kann. Nur wenn alle Individuen ausreichend investierten, kann das öffentliche Gut zur Verfügung gestellt werden. Beispiele dafür finden sich in menschlichen Gesellschaften in unterschiedlichen Bereichen wie beispielsweise in Form von Krankenversicherungen, Infrastruktur oder des öffentlichen Verkehrs. Hierbei besteht für den Einzelnen ein Anreiz zum Trittbrettfahren, d.h. es wird kein Beitrag zur Bereitstellung des öffentlichen Gutes geleistet aber da einzelne Individuen nur schwer von der Nutzung des öffentlichen Gutes ausgeschlossen werden können, ist man individuell incentiviert sich nicht zu beteiligen (so ist es beispielsweise verhältnismäßig schwierig den Zugang zu einer Straße zu kontrollieren). Allmendegut-Spiele unterscheiden sich davon dadurch, dass es eine Ressource gibt, die von allen Individuen gemeinschaftlich genutzt werden kann. Allerdings besteht bezüglich der Nutzung dieser Ressource Rivalität zwischen den Individuen, da der Gesamtnutzen der Ressource durch die individuelle Nutzung abnimmt. Das Dilemma entsteht nun durch Überbeanspruchung der Ressource, da die Kosten für die Nutzung der Ressource von allen Individuen getragen werden, während der Nutzen nur individuell wahrgenommen wird. Beispiel hierfür sind die gemeinschaftliche Nutzung natürlicher Ressourcen.

2.1.4 Sequentielle Soziale Dilemmas

Um die oftmals abstrakten Modelle sozialer Dilemmas näher an reale Probleme heranzuführen, wurde eine Erweiterung des Konzepts mit dem Namen sequentielle soziale Dilemmas (SSDs) vorgeschlagen [45]. SSDs bieten eine zeitliche

Erweiterung, so dass die Interaktion zwischen den Agenten länger andauert. Agenten lernen daher kooperatives bzw. nicht-kooperatives Verhalten in Form einer einzelnen Aktion, sondern in Form von zustandsabhängigen Strategien, so dass ein Agent mal kooperativ mal weniger kooperativ sein kann. Eine weitere Motivation für SSDs ist die Beobachtung, dass es in den meisten Echtwelt-Situationen verschiedene Abstufungen kooperativen Verhaltens geben kann und dass Individuen den Grad ihrer Kooperationsbereitschaft möglicherweise von Zeit zu Zeit ändern. Auch die Notwendigkeit, dass Aktionen simultan getroffen werden müssen, ist durch die zeitliche Erweiterung in SSDs aufgelöst. Dies kann für das Entstehen von kooperativem Verhalten wichtig sein, da die Information darüber, ob andere Individuen beginnen zu kooperieren für die eigene Entscheidung zur Kooperation eine Rolle spielen kann.

2.1.5 Verhandlungsspiele

Neben den beschriebenen sozialen Dilemmas gibt es eine weitere Klasse an Szenarien, die Untersuchungsgegenstand der Spieltheorie sind. Diese Spiele werden als Verhandlungssituationen bezeichnet. Verhandlungssituationen gehören zu den grundlegendsten wirtschaftlichen Aktivitäten, daher hat deren Untersuchung eine hohe praktische Relevanz. Der Unterschied zu sozialen Dilemmas ist, dass bei Verhandlungen eine Zusammenarbeit generell von allen Beteiligten gewünscht wird, es jedoch unterschiedliche Präferenzen bezüglich der genauen Form der Zusammenarbeit gibt [59, 72]. Die Definition einer Verhandlungssituation im spieltheoretischen Sinne sieht vor, dass kein Mitglied der verhandelnden Individuen zu einem bestimmten Verhandlungsergebnis gezwungen werden kann, d.h. jedes Mitglied kann von einem Veto Gebrauch machen. Macht ein Verhandlungsteilnehmer von seinem Veto-Recht Gebrauch, so wird für alle Teilnehmer ein vordefiniertes Ergebnis realisiert, welches das Scheitern der Verhandlung repräsentiert (sogenanntes status-quo Ergebnis). Generell gilt, dass kein Mitglied eine gescheiterte Verhandlung gegenüber allen erfolgreichen Verhandlungsmöglichkeiten bevorzugt, so dass eine Kollaboration von allen Mitgliedern generell gewünscht wird.

Eine Verhandlungssituation zwischen zwei Spielern ist in Abbildung 2.3 dargestellt, wobei der graue Bereich die Menge der realisierbaren Ergebnisse darstellt. In diesem Fall möchte Spieler 1, dass der tatsächliche Einigungspunkt so weit rechts wie möglich liegt, während Spieler 2 Ergebnisse präferiert, die einen größeren y-Wert haben. Das tatsächlich realisierte Ergebnis ist dann Gegenstand der Verhandlung zwischen den Teilnehmern. Wichtig ist, dass jeder Teilnehmer die Möglichkeit hat, gegen jede Vereinbarung ein Veto einzulegen.

Im spieltheoretischen Kontext basiert die Analyse von Verhandlungen in der Regel wieder auf der Annahme vollkommen rationaler Akteure. Nash [59] stellt ein axiomatisches Modell vor, das es erlaubt, eine eindeutige Lösung für das Problem der Aufteilung eines gemeinsamen Gutes zwischen zwei rationalen Verhandlungspartnern abzuleiten. Dieses Modell einer Verhandlungssitua-

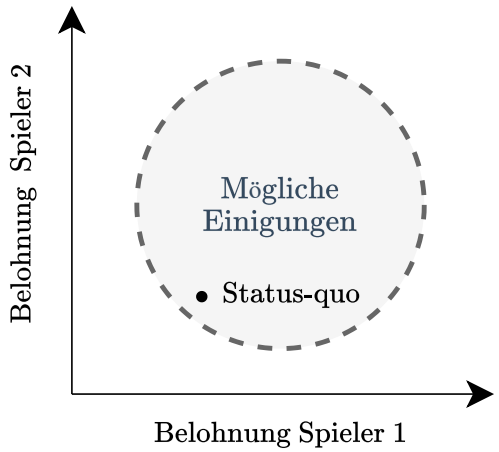


Abbildung 2.3: Verhandlungssituation zwischen zwei Spielern. Der Kreis markiert die Möglichkeiten zur Zusammenarbeit, doch es gibt Interessenskonflikte.

tion wird daher auch als Nash-Verhandlung bezeichnet und ist formal definiert durch eine Menge von Verhandlungspartnern $\mathcal{N}$, die Verhandlungsmenge A, die Uneinigkeitsmenge D und für jeden Verhandlungspartner $i \in \mathcal{N}$ eine Nutzenfunktion $u_i : A \cup D \to \mathbb{R}$, die bis auf positiv affine Transformationen der Funktion eindeutig ist. Die Menge der möglichen Vereinbarungen, die Uneinigkeitsmenge und die Nutzenfunktionen reichen aus, um die Menge S aller Nutzenpaare zu konstruieren, die mögliche Ergebnisse der Verhandlung sind, d.h. $S = \{(u_1(a), ..., u_n(a))\}$, und den eindeutigen Punkt der Uneinigkeitsmenge $d = \{u_1(d), ..., u_n(d)\}$. Das Paar (S, d) wird dann als ein Verhandlungsproblem bezeichnet und bildet das Modell für die axiomatische Lösung nach Nash [59]. Die Menge aller Verhandlungsprobleme wird mit B bezeichnet. Eine Verhandlungslösung ist eine Funktion $f : B \to \mathbb{R}^2$, die jedem Verhandlungsproblem $(S, d) \in B$ ein eindeutiges Element von S zuordnet. Nash stellte vier Axiome auf, die für eine Verhandlungslösung gelten müssen [59]:

1. *Pareto-Effizienz*: Die Spieler werden sich niemals auf ein Ergebnis einigen, wenn es ein anderes Ergebnis gibt, bei dem beide Spieler bessergestellt sind.

2. *Invarianz zu äquivalenten Nutzendarstellungen*: Das Verhandlungsergebnis ist invariant, wenn die Nutzenfunktion und der status-quo Punkt durch eine lineare Transformation skaliert werden.

3. *Symmetrie*: Es wird angenommen, dass alle Verhandlungspartner, die gleichen Verhandlungsfähigkeiten besitzen, d.h. alle Asymmetrien zwischen den Akteuren müssen bereits in (S, d) modelliert sein. Daher sollte eine symmetrische Verhandlungssituation zu gleichen Ergebnissen für alle Spieler führen.

4. *Unabhängigkeit von irrelevanten Alternativen*: Wenn eine für eine Verhandlungssituation gefundene Lösung einer kleineren Teilmenge zugeordnet werden kann, dann ist die Lösung dieselbe, wenn die neue machbare Menge auf diese Teilmenge reduziert wird.

Nash [59] konnte zeigen, dass es unter diesen Axiomen eine eindeutige Verhandlungslösung $f : B \to \mathbb{R}$ gibt, die definiert ist durch:

$$f(S, d) = \underset{(d_1,d_2)\leq(s_1,s_2\in S)}{\arg\max} (s_1 - d_1) \times (s_2 - d_2)$$

Im Gegensatz zu dem axiomatischen Modell von Nash gibt es auch andere Modellierungsansätze aus den Bereichen der evolutionären Spieltheorie [25], der genetischen Algorithmen [26] und der neuronalen Netze [65]. Dabei wird die Annahme vollkommener Rationalität fallen gelassen und durch begrenzt rationale Agenten ersetzt, um die Entwicklung sich entwickelnder Strategien zu analysieren.

2.2 Reinforcement Learning

Reinforcement Learning (RL) bezeichnet einen Teilbereich des maschinellen Lernens, bei dem ein autonomer Agent versucht, mittels eines Trial-and-Error geleiteten Prozesses eine optimale Verhaltensvorschrift oder Strategie zu erlernen. In den letzten Jahren konnten mit Methoden des Reinforcement Learning zahlreiche Durchbrüche in verschiedenen Bereichen erzielt werden, darunter Brettspiele wie Go [83], Echtzeit-Videospiele [88] oder zur Erzeugung komplexer Verhaltensweisen in kooperativen Multiagentensystemen [36]. Im Gegensatz zu anderen Techniken des maschinellen Lernens, wie beispielsweise dem überwachten Lernen, liegen zu Beginn des Trainings keine gelabelten Eingangsdaten vor, d.h. der Agent kann nicht anhand bereits gekennzeichneter Trainings-Beispiele lernen. Dem Agenten wird also lediglich kommuniziert, dass er ein bestimmtes Ziel erreichen soll, nicht jedoch wie er es erreichen kann. Das zu erreichende Ziel wird über eine Belohnungsfunktion definiert, die dem Agenten ein numerisches Signal in Folge seiner ausgeführten Handlungen zurückgibt. Der Agent muss über den Verlauf mehrerer Episoden selbst ausloten, welche Strategien erfolgreich in Bezug auf die erhaltenen Belohnungen sind. Bezogen auf den Trainingsprozess ergeben sich dadurch verschiedene Schwierigkeiten, da die Trainingsdaten, die der Agent erzeugt nicht unkorreliert sind, sondern gemäß der aktuellen Explorationsstrategie des Agenten generiert werden und somit starke sequentielle Korrelationen aufweisen.

Zentrale Komponenten des Reinforcement Learning sind der Agent und die ihm zunächst unbekannte Umgebung, für die der Agent versucht eine optimale Strategie zu lernen. Die Interaktion des Agenten mit der Umgebung ist schematisch in Grafik 2.4 dargestellt. Die Umgebung befindet sich zum Zeitpunkt t in einem Zustand s_t. Basierend auf seiner Beobachtung $\mathcal{O}(s_t)$ des Zustands

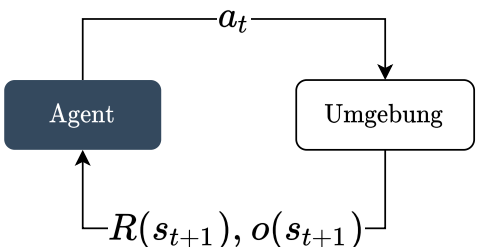


Abbildung 2.4: Beim Reinforcement Learning interagiert der Agent wiederholt mit einer unbekannten Umgebung.

s_t wählt der Agent eine Aktion a_t aus, die nun ausgeführt wird. Daraufhin geht die Umgebung in einen neuen Zustand s_{t+1} über, und der Agent erhält eine Beobachtung des Zustands $\mathcal{O}(s_{t+1})$ sowie eine Belohnung $R(s_{t+1})$. Ziel des Agenten ist es, durch die wiederholte Interaktion eine optimale Strategie zu erlernen. Diese Strategie wird als *Policy* π bezeichnet. Handelt es sich nicht um eine stochastische Policy, so ist π eine Wahrscheinlichkeitsverteilung, so dass $\pi(a|s)$ die Wahrscheinlichkeit angibt, dass der Agent in Zustand s die Aktion a wählt.

2.2.1 Markov-Entscheidungsprozess

Die Entwicklung der Umgebung in Folge der Aktionswahl des Agenten lässt sich formal als Markov-Entscheidungsprozess (MEP) beschreiben, der eine grundlegende Abstraktion sequentieller Lernprobleme darstellt [85]. Dabei wird das Problem auf drei relevante Signale reduziert: Die Aktionen, mit denen der Agent auf die Umgebung einwirkt, die Zustände, die dem Agenten die Entscheidungsgrundlage für die Aktionswahl geben und die Belohnungen durch die der Agent lernt welche Aktionen hilfreich bei der Zielerreichung sind. Formal ist ein Markov-Entscheidungsprozess definiert als ein Tupel $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, wobei $\mathcal{S}$ eine endliche Menge an Zuständen beschreibt, $\mathcal{A}$ stellt die endliche Menge an Aktionen des Agenten dar, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to PD(s)$ ist die Zustandstransitionsfunktion, die für einen gegeben Zustand s und eine Aktion a die Wahrscheinlichkeitsverteilung über die Folgezustände angibt, sowie die Belohnungsfunktion $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, die für einen gegeben Zustand s und eine Aktion a die numerische Belohnung des Agenten angibt.

Ziel des Agenten ist es, die Summe der zu erwartenden, diskontierten Belohnungen zu maximieren. Diese Summe wird als Erlös (engl. *Return*) bezeichnet und ist zum Zeitpunkt t definiert als:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

wobei $0 \geq \gamma \geq 1$ einen Diskontierungsfaktor darstellt. Durch die Diskontierung werden weiter in der Zukunft liegende Belohnungen gegenüber zeitlich

näher liegenden, niedriger gewichtet, so dass der Agent frühere Belohnungen gegenüber späteren Belohnungen bevorzugt, da eine Belohnung die k Schritte in der Zukunft liegt, lediglich das γ^{k-1}-fache Wert ist im Vergleich zu einer Belohnung, die unmittelbar erhalten wird. Je kleiner γ, desto *kurzsichtiger* optimiert der Agent seinen Erlös, da zukünftige Belohnungen weniger gewichtet werden. Je größer γ gewählt wird, desto *weitsichtiger* berücksichtigt der Agent auch zukünftige Belohnungen.

2.2.2 Werte-Funktionen

Zum Erlernen einer Policy gehen viele Algorithmen des Reinforcement Learning den Weg über das Erlernen einer Wertefunktion (engl. *Value-Functions*). Etwas informell lässt sich sagen, dass eine Wertefunktion dem Agenten Auskunft darüber gibt, wie gut es ist, in einem bestimmten Zustand zu sein, bzw. wie gut es ist in einem bestimmten Zustand eine bestimmte Aktion zu wählen. Höhere Werte der Wertefunktion sind mit höheren erwarteten zukünftigen Belohnungen assoziiert, so dass der Agent die Wertefunktion verwenden kann, um Aktionen zu wählen, die höhere Belohnungen versprechen. Da die zukünftigen zu erwartenden Belohnungen davon abhängen, welcher Policy der Agent zukünftig folgen wird, werden Wertefunktionen immer bezüglich einer bestimmten Policy definiert. Der Wert (engl. *Value*) eines Zustands s zu einer gegebenen Policy π wird mit $v_\pi(s)$ bezeichnet und ist die erwartete Belohnung, wenn der Agent in Zustand s startet und Aktionen gemäß der Policy π wählt [85]:

$$v_\pi(s) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1|S_t=s}\Big]$$

Die Funktion $v_\pi(s)$ gibt den Wert des Zustands s an und wird daher als Zustandswertefunktion (engl. *State-Value Function*) bezeichnet. Analog lässt sich die Wertefunktion definieren, die den erwarteten zukünftigen Erlös angibt, wenn der Agent in Zustand s Aktion a wählt und anschließend Policy π folgt [85]:

$$q_\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1|S_t=s,A_t=a}\Big]$$

Die Funktion $q_\pi(s,a)$ gibt also den Wert basierend auf Zustand s und Aktion a zurück und wird daher als Zustand-Aktions-Wertefunktion (engl. *State-Action Value Function*) bezeichnet.

2.2.3 Bellman-Gleichung

In einem Markov-Entscheidungsprozess führt die Wahl einer Aktion in einem Zustand zu einem Übertritt in einen neuen Zustand. Durch diese sequentielle Verbindung von Zuständen lassen sich rekursive Eigenschaften für die Wertefunktionen definieren. Insbesondere gibt es eine rekursive Beziehung zwischen

dem Wert eines Zustands und den möglichen Folgezuständen. So gilt für den Wert $v_\pi(s)$ eines Zustand s unter Policy π und den möglichen Folgezuständen s' [85]:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

wobei $p(s', r|s, a)$ die Wahrscheinlichkeit angibt, in Zustand s durch Wahl von Aktion a in Zustand s' zu kommen und eine Belohnung r zu erhalten. Diese Beziehung zwischen dem Wert eines Zustands und den möglichen Folgezuständen wird als Bellman-Gleichung bezeichnet. Viele Algorithmen beruhen auf der Anwendung der Bellman-Gleichung, indem diese Gleichung als Aktualisierungsregel verwendet wird.

Eine Policy π^* wird als optimal bezeichnet, wenn sie einen höheren erwarteten Erlös erzielt als alle anderen Policies. Es kann allerdings mehrere optimale Policies geben, daher ist π^* optimal, wenn es keine Policy gibt, die einen höheren Erlös erzielt. Für die Menge der optimalen Policies gilt, dass alle dieselbe optimale Wertefunktion v^* haben. Für die optimale Wertefunktion v^* gilt [85]:

$$v^*(s) = \max_\pi v_\pi(s) \forall s \in \mathcal{S}$$

Analog lässt sich die optimale Zustands-Aktions-Wertefunktion q^* definieren [85]:

$$q^*(s, a) = \max_\pi q_\pi(s, a) \forall s \in \mathcal{S}$$

Die Bellman-Optimalitäts-Gleichung besagt nun, dass der Wert eines Zustands s unter einer optimalen Policy π^* gleich dem erwarteten Erlös nach der Wahl der besten Aktion in diesem Zustand ist [85]:

$$v^*(s) = \max_a q_{\pi^*}(s, a) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v^*(s')]$$

Die Bellman-Optimalitäts-Gleichung stellt also die Verbindung der optimalen Zustands-Wertefunktion mit der optimalen Zustands-Aktions-Wertefunktion her.

2.2.4 Wertebasierte Methoden

Methoden des Reinforcement Learning lassen sich grob in *wertebasierte-* und *Policy-basierte* Methoden unterteilen. Bei wertebasierten Methoden wird zunächst eine Wertefunktion erlernt, die dann dazu verwendet wird, Aktionen über deren Wert abzuleiten. Ist beispielsweise die optimale Zustands-Aktions-Wertefunktion bekannt, so lässt sich daraus eine optimale Policy ableiten, indem über die argmax-Operation in jedem Zustand die Aktion mit dem höchsten assoziierten Wert gewählt wird. Policy-basierte Methoden gehen nicht den Weg über das Erlernen einer Wertefunktion, sondern suchen direkt nach einer optimalen Policy im Raum der möglichen Policies. Der nächste Abschnitt

beschreibt zunächst eines der gängigsten wertebasierten Verfahren, das *Q-Learning* genannt wird.

2.2.4.1 Q-Learning

Einer der bekanntesten Algorithmen zum Erlernen optimaler Policies wird als *Q-Learning* bezeichnet [92]. Q-Learning beruht auf der Technik des *Temporal-Difference-Learning*, das heißt, die Werte der Q-Funktion für Zustands-Aktions-Paare werden, basierend auf den Werten nachfolgender Zustands-Aktions-Paare gemäß der Bellman-Gleichung, aktualisiert. Ein großer Vorteil des Q-Learning besteht darin, dass für die Anwendung lediglich Daten in Form von generierten Erfahrungstupeln benötigt werden. Es ist also keinerlei Wissen über die Transitionswahrscheinlichkeiten der Zustände nötig. Das Generieren der zum Training notwendigen Erfahrungen läuft dabei parallel zum Erlernen der optimalen Policy. Durch die Interaktion mit der Umgebung erhält der Agent zum Zeitschritt t ein Erfahrungs-Tupel (s_t, a_t, r_r, s_{t+1}), bestehend aus einem Zustand s_t, einer Aktion a_t, einer Belohnung r_t, sowie einem Folgezustand s_{t+1}. Zum Beginn des Trainings wird die Q-Funktion des Agenten zufällig initialisiert. Nachdem Erfahrungen generiert wurden, wird die Q-Funktion dann gemäß folgender Regel aktualisiert:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a'} Q(s', a') - Q(s_t, a_t)]$$

Der rechte Teil der Gleichung ist eine gewichtete Summe, bestehend aus dem bisherigen Wert der Funktion an der Stelle $Q(s_t, a_t)$ und dem sogenannten Target $r_t + \gamma \max_{a'} Q(s', a')$, das mit dem Faktor $0 < \alpha \leq 1$ gewichtet wird. α wird daher auch als Lernrate bezeichnet. Bei dem Target handelt es sich um eine Schätzung des zu erwartenden Erlöses, basierend auf den bisher erworbenen Erfahrungen. Der Term $r_t + \gamma \max_{a'} Q(s', a') - Q(s_t, a_t)$ ist der geschätzte Fehler zwischen dem zu erwartenden Erlös, basierend auf der beobachteten Belohnung r_t und dem besten Wert des Folgezustandes $\max_{a'} Q(s', a')$ und dem bisher geschätzten zu erwartenden Erlös $Q(s_t, a_t)$.

Q-Learning ist ein sogenannter *Qff-Policy*-Algorithmus, wobei mit *Off-Policy* das Verfahren beschrieben wird unter dem eine Policy zum Generieren der Erfahrungstupel verwendet wird (Verhaltenspolicy), jedoch eine optimale Policy Policy gelernt werden soll (Zielpolicy), die allerdings nicht zum Erzeugen von Trainingsdaten verwendet wird. Diese Trennung erlaubt es eine Verhaltenspolicy zu definieren, die den Exploration-Exploitation-Trade-Off (siehe Abschnitt 2.2.8) effizient steuert. Dazu wird häufig das *Epsilon-Greedy* Verfahren verwendet. Dabei werden Aktionen gemäß folgender Vorschrift gewählt: bei jeder Aktionswahl wird mit einer Wahrscheinlichkeit von ϵ eine zufällige Aktion gewählt, wodurch sicher gestellt wird, dass alle möglichen Aktionen in einem Zustand erkundet werden (Exploration). Mit der Gegenwahrscheinlichkeit $1 - \epsilon$ wird die beste Aktion gemäß der aktuellen Q-Funktion gewählt, dadurch wird sicher gestellt, dass bereits erworbenes Wissen auch genutzt wird

(Exploitation). Der Wert von ϵ kann dabei über den Verlauf des Trainings dekrementiert werden, so dass zunehmend nur noch optimale Aktionen gewählt werden und der Agent weniger zufällige Aktionen wählt.

2.2.5 Approximative Methoden

Viele interessante Anwendungsfälle für Methoden des Reinforcement Learning haben einen sehr großen Zustandsraum $\mathcal{S}$. In diesen Fällen können die Werte der Wertefunktionen nicht mehr für alle Zustände gespeichert werden. Darüber hinaus ergibt sich für große Zustandsräume das Problem, dass es in endlicher Zeit nicht möglich ist, alle Zustände zu explorieren. Daher ist es für den Agenten notwendig zu generalisieren, also Techniken anzuwenden, die es ermöglichen, von Zuständen, die bereits gesehen wurden, auf ähnliche aber bisher ungesehene Zustände zu schließen. Beide Probleme können mit parametrisierten Funktionsapproximatoren angegangen werden. Der Vorteil liegt darin, dass die Anzahl der zu trainierenden Parameter deutlich niedriger ist als die Anzahl der Zustände und somit auch bei großen Zustandsräumen ein effizientes Training der Modelle möglich ist. Zu den am häufigsten eingesetzten Funktionsapproximatoren gehören künstliche neuronale Netze. Ein neuronales Netz kann beispielsweise zur Repräsentation der Q-Funktion für das Q-Learning verwendet werden. Durch die Integration neuronaler Netze in Methoden des Reinforcement Learning wird ein weiterer Trainingsprozess notwendig, der auf Techniken des überwachten Lernens beruht. Dadurch können sich eigene Schwierigkeiten für den Trainingsprozess des RL Agenten ergeben, wie Konvergenzprobleme oder Nicht-Stationarität, die das Training erschweren können. Im Folgenden wird daher zunächst auf das Training neuronaler Netze eingegangen, bevor der Algorithmus *Deep-Q-Networks* [57] vorgestellt wird, eine Variante von Q-Learning, bei der die Zustands-Aktions-Wertefunktion in Form eines tiefen neuronalen Netzes repräsentiert wird.

2.2.5.1 Neuronale Netze

Zu den am häufigsten eingesetzten nicht-linearen Funktionsapproximatoren im Reinforcement Learning gehören künstliche neuronale Netze. Ein künstliches neuronales Netz stellt eine Struktur miteinander verbundener Einheiten, sogenannter Neuronen dar, die ein gegebenes Eingangssignal durch die Bildung von linear-Kombinationen und anschließender Anwendung von nicht-linearen Aktvierungsfunktionen verarbeiten und weiterleiten. Ein neuronales Netz setzt sich dabei aus verschiedenen Schichten zusammen: Das Eingangssignal bildet die erste Schicht, worauf beliebig viele sogenannte *versteckte* Schichten folgen. Die letzte Schicht wird als Ausgangsschicht bezeichnet. Grafik 2.5 zeigt den schematischen Aufbau eines Neuronalen Netzes mit einer Eingangsschicht, zwei versteckten Schichten und einer Ausgangsschicht. Die Kreise stellen die Einheiten dar, welche die linear-Kombinationen der daran eingehenden Signale bilden und anschließend eine (nicht-lineare) Aktivierungsfunktion anwenden.

Das entstehende Ausgangssignal wird an die nächste Einheit weitergegeben. Netze mit mehr als einer versteckten Schicht werden als *tiefe* neuronale Netze bezeichnet. Neuronale Netze, die Eingangssignale lediglich vorwärtsgerichtet weiterleiten, werden als *Feedforward*-Netze bezeichnet (siehe Grafik 2.5). Netze, die zirkuläre Strukturen enthalten, also Signale auch rückleiten können, werden als Rekurrente-Netze bezeichnet.

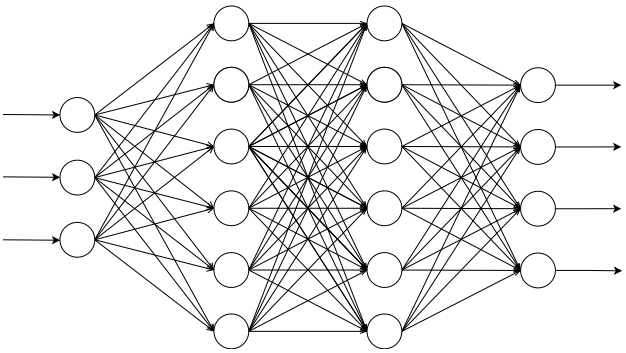


Abbildung 2.5: Schematischer Aufbau eines *feedforward*-neuronalen Netzes mit zwei versteckten Schichten, drei Eingangssignalen und vier Ausgangssignalen.

Ein Feedforward-Netz lässt sich als Verkettung einzelner Funktionen verstehen, beispielsweise könnte ein neuronales Netz eine Funktion $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$ repräsentieren, die sich aus den Funktionen $f^{(3)}, f^{(2)}$ und $f^{(1)}$ zusammensetzt. Ziel des Trainings eines neuronalen Netzes ist es, eine Zielfunktion f^* zu approximieren. Beispielsweise könnte die Zielfunktion ein Klassifikator sein, der einem Eingabevektor $\mathbf{x}$ eine Kategorie zuweist. Ein Feedforward-Netz $y(\mathbf{x}; \theta)$ approximiert eine Zielfunktion, indem es eine Belegung von Parametern θ erlernt, die die Zielfunktion möglichst gut annähern.

Funktional lässt sich ein Feedforward-Netz folgendermaßen darstellen:

$$y(\mathbf{x}, \theta) = f(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x}))$$

wobei θ den Vektor aus (zu trainierenden) Gewichten beschreibt, $f(\cdot)$ ist eine nicht-lineare Aktivierungsfunktion, w_j sind die Elemente von θ und ϕ_j sind die potentiell ebenfalls nicht-linearen Ausgangssignale vorausgehender Schichten des neuronalen Netzes. Das neuronales Netz beschreibt also eine Serie funktionaler Transformationen: Aus den Eingangsvariablen $x_1, ..., x_D$ wird zuerst eine linear-Kombination gebildet:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

wobei $j = 1, ..., M$ die Anzahl der Ausgangssginale der ersten Schicht ist, und

D die Dimension des Eingangs-Datenraums. Die (1) in $w_{ij}^{(1)}$ gibt an, dass es sich um die Gewichte der ersten Schicht handelt. Die sogenannten Aktivierungen a_j werden durch eine differenzierbare, nicht-lineare Aktivierungsfunktion h weiter transformiert:

$$z_j = h(a_j)$$

Diese sogenannten versteckten Einheiten werden ihrerseits wieder in der nächsten Schicht in Form von linear-Kombinationen zusammengefasst:

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

wobei die (2) angibt, dass es sich nun um die Gewichte der zweiten Schicht handelt für $k = 1, ..., K$, so dass K die Gesamtzahl der Ausgangssignale der zweiten Schicht darstellt. Die Aktivierungen a_k werden ebenfalls wieder mittels Aktivierungsfunktionen transformiert.

2.2.5.1.1 Aktivierungsfunktionen Wie oben beschrieben basiert die Funktionsweise eines Feed-Forward-Netzes auf der Berechnung einer affinen Transformation $a_j = \sum_{i=1}^{D} w_{ji}^{(k)} x_i + w_{j0}^{(k)}$ zum Eingangsvektor $\mathbf{x}$ sowie der anschließenden Anwendung einer nicht-linearen Aktivierungsfunktion $z_j = h(a_j)$ zur Berechnung des Ausgangssignals der Einheit j in Schicht k. Erst durch die Anwendung der Funktion h wird das Ausgangssignal in eine nicht-lineare Form transformiert. Die Wahl der Aktivierungsfunktion h hängt dabei von verschiedenen Entscheidungen ab und ist nach wie vor aktiver Gegenstand aktueller Forschung [29]. Eine der am häufigsten verwendeten Aktivierungsfunktionen sind *Rectified Linear Units* (ReLU), bei denen als Aktivierungsfunktion $h = \max\{0, z\}$ verwendet wird. Generell richtet sich die Wahl der Aktivierungsfunktion nach der Art der zu lernenden Daten und der angedachten Zielverteilung der Daten. Handelt es sich bei den Aktivierungen a_k um die letzte Schicht des neuronalen Netzes, so kann für Regressionsprobleme die Identitätsfunktion zur Aktivierung verwendet werden, also $y_k = a_k$ oder im Falle einer Multiklassen-Klassifikation die Softmax-Aktivierungsfunktion.

2.2.5.2 Training Neuronaler Netze

Wird ein neuronales Netz mit Gewichten θ zur Approximation einer Zielfunktion $\mathbf{y} = f^*(\mathbf{x})$ verwendet, müssen die Gewichte θ durch entsprechendes Training gelernt werden. Obwohl die Elemente von θ für Funktionen mit kleinem Definitionsbereich (sowie beispielsweise die XOR-Funktion) durch manuelles Ausprobieren gefunden werden können, ist das für die meisten interessanten Szenarien nicht möglich. Zum Erlernen der Gewichte werden daher in der Regel (stochastische) Gradienten-Verfahren verwendet, wobei der Gradient dazu verwendet wird die Gewichte gemäß des Gradienten-Signals zu verändern. Dazu ist es notwendig, zunächst eine zu optimierende Zielfunktion $J(\theta)$ zu

definieren. $J(\theta)$ wird als Verlustfunktion bezeichnet und in Abhängigkeit des zugrundeliegenden Lernproblems gewählt. Im Gegensatz zu linearen Modellen, ist die Verlustfunktion durch die Integration nicht-linearer Aktivierungsfunktionen in den meisten Fällen nicht konvex, wodurch der Einsatz von iterativen, gradientenbasierten Optimierungsverfahren notwendig wird, die keine globalen Konvergenz-Garantien besitzen und stark von der Wahl der initialen Gewichte abhängen [29].

In den meisten Fällen wird die Verlustfunktion nach dem Prinzip der *Maximum-Likelihood* gewählt, so dass der Verlust als die *Cross-Entropy* zwischen der Verteilung der Trainings-Daten und der Verteilung, die durch das gewählte Modell definiert wird, beschrieben werden kann. Die spezielle Form der Verlustfunktion ändert sich daher von Modell zu Modell, je nach Wahl der Modell-Verteilung. Außerdem hängt die Wahl der Verlustfunktion von der Wahl der Ausgangseinheiten zusammen.

Handelt es sich bei der gewünschten Modell-Verteilung beispielsweise um eine Bernoulli-Verteilung, also versucht man eine Wahrscheinlichkeit über den Ausgang eines Experiments mit zwei möglichen Ausgängen zu modellieren, so entspricht das Ausgangssignal des neuronalen Netzes der Wahrscheinlichkeit $P(y = 1|x)$. Damit es sich um eine valide Wahrscheinlichkeit handelt, muss gelten, dass $P \in [0, 1]$. Um diese Bedingung zu erfüllen, muss das Ausgangssignal entsprechend transformiert werden. Obwohl in der Theorie eine künstliche Begrenzung durch die Schranken von 0 und 1 durch die Wahl $P(y = 1|\mathbf{x}) = max\{0, min\{1, \mathbf{w}^T\mathbf{h} + b\}\}$ realisiert werden kann, bietet sich dennoch eine andere Wahl der Modellierung von $P(y = 1|x)$ an. Besser geeignet für die Modellierung einer Bernoulli-Verteilung ist die Wahl einer Sigmoid-Funktion zur Transformation des Ausgangsignals in den Wertebereich $[0, 1]$, so dass

$$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{h} + b),$$

wobei σ die Funktion $\sigma(x) = \frac{1}{1+exp(-x)}$ darstellt. Durch die Wahl der Sigmoid-Funktion ist sichergestellt, dass die zu minimierende Verlust-Funktion während des Trainings immer ein Gradientensignal aufweist, das ungleich Null ist und somit Auskunft darüber gibt in welche Richtung die Gewichte geändert werden müssen, um den Verlust zu verringern. Darüber hinaus liegen die Werte immer zwischen 0 und 1, wodurch sichergestellt ist, dass es sich um die Modellierung einer Wahrscheinlichkeit handelt. Im Falle einer kategorischen Zielverteilung, also der Modellierung einer diskreten Variablen mit n möglichen Ausprägungen, kann die *Softmax*-Funktion verwendet werden, die formal definiert ist als:

$$\text{softmax}(\mathbf{z})_i = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

wobei $\mathbf{z} = \mathbf{W}^T\mathbf{h} + \mathbf{b}$.

2.2.5.3 Deep-Q-Networks

Nachdem die generelle Funktionsweise und das Training neuronaler Netze in den vorangehenden Abschnitten erläutert wurden, wird in diesem Abschnitt der Algorithmus *Deep-Q-Networks* (DQN) [56] beschrieben, bei dem die Q-Funktion des Agenten durch ein tiefes neuronales Netz repräsentiert wird. DQN war einer der ersten großen Meilensteine bei dem Versuch Reinforcement Learning und neuronale Netze für den Einsatz in Domänen mit großem Zustandsräumen zu kombinieren. Durch den Einsatz eines neuronalen Netzes für die Q-Funktion ist es möglich Policies für Domänen mit sehr großen Zustandsräumen, wie beispielsweise Atari-Umgebungen zu trainieren. Obwohl es bereits früher Ansätze dazu gab, wurden in [56] verschiedene Neuerungen eingeführt, die das stabile Training des neuronalen Netzes ermöglichten: Die hohe Korrelation der Daten, die zur Instabilität des Trainings beitragen kann, wird durch die Methode des *Experience-Replay* abgemildert. Dabei werden die Daten zu einem Trainingsschritt zufällig aus einem Puffer-Speicher gezogen, so dass das Netz pro Schritt Erfahrungstupel aus unterschiedlichen Episoden und Phasen des Trainings bekommt und somit die starken Korrelationen der Beobachtungen reduziert. Darüber hinaus werden anstatt eines einzelnen Netzes zur Abbildung der Q-Funktion zwei Netze verwendet, ein *Policy*-Netz und ein *Target*-Netz. Das Policy-Netz wird verwendet, um die Aktionen des Agenten zu kontrollieren und Erfahrungsdaten zu generieren. Das Target-Netz wird zur Aktualisierung des Policy-Netzes während der Trainingsschritte verwendet, also zur Vorhersage der Q-Werte für nachfolgende Zustands-Aktionspaare. Das Target-Netz wird periodisch mit dem Policy-Netz aktualisiert, wodurch zeitliche Korrelationen mit dem Update-Target reduziert werden. Durch die Trennung der Q-Funktion in zwei Netze wird erreicht, dass nicht das Netz mit Werten aktualisiert wird, die es selbst erzeugt hat, wodurch das Training erheblich stabilisiert werden kann. Das Training des Q-Netzwerks basiert auf der Minimierung der Verlustfunktion $\mathcal{L}(\theta)$ [56]:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s' \sim p(\cdot)}[(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i))^2]$$

wobei θ_{i-1} die Gewichte des *Target*-Netzes und θ_i die Gewichte des *Policy*-Netzes kennzeichnet.

Die Original-Implementierung von DQN beruht auf der Verwendung eines *convolutional neural network* (CNN), einer speziellen Netz-Architektur, die basierend auf Faltungen und Vereinigungen die räumlichen Korrelationen des Eingabevektors ausnutzt. CNNs sind daher insbesondere für die Verarbeitung von Bilddaten geeignet. Durch die Verwendung eines CNNs kann DQN beispielsweise direkt auf den rohen Pixel-Daten eines Atari-Emulators verwendet werden. Da in den vorgestellten Experimenten dieser Arbeit primär auf relativ niedrigdimensionalen Daten trainiert wird, wurde auf den Einsatz von CNNs verzichtet, da diese oftmals mit einer verlängerten Trainingszeit einhergehen.

2.2.6 Gradientenbasierte Methoden

Wie im Abschnitt 2.2.4 beschrieben, beruhen viele Algorithmen des Reinforcement Learning auf dem Erlernen einer Werte-Funktion. Daneben gibt es eine zweite Klasse an Algorithmen, die statt einer Werte-Funktion direkt nach einer optimalen Policy im Raum der Policies suchen [85]. Eine Policy π lässt sich dabei als parametrisierte Funktion darstellen, wobei $\theta \in \mathbb{R}^n$ den Parameter-Vektor beschreibt. $\pi(a|s, \theta)$ entspricht der Wahrscheinlichkeit im Zustand s Aktion a zu wählen unter dem Parameter-Vektor θ. Es wird also nach einer Belegung der Parameter θ (auch Gewichte genannt) gesucht, die eine optimale Policy π^* definiert. Für die Suche nach den optimalen Gewichten wird ein Performance-Maß $J(\theta)$ definiert, dessen Gradient dann für die iterative Verbesserung der Policy verwendet werden kann. Die Parameter werden dann gemäß folgender Regel aktualisiert [85]:

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t)$$

Methoden, die darüber hinaus auch noch eine Werte-Funktion lernen, um zusätzliches Feedback zum Aktualisieren der Policy zu gewinnen, werden als *Actor-Critic*-Methoden bezeichnet.

2.2.6.1 Proximal-Policy Optimization

Ein Algorithmus der der Klasse der *Actor-Critic*-Methoden zugeschrieben werden kann, ist *Proximal Policy Optimization* (PPO) [80]. Eine generelle Schwierigkeit bei gradientenbasierten Verfahren, liegt in der Sensitivität der Verfahren in Bezug auf die gewählte Schrittgröße der Anpassungen der Gewichte [80]. Einige Methoden regulieren daher die Stärke der Aktualisierungen, so dass die Policy-Updates einen gewissen Schwellenwert nicht überschreiten [79]. Der Algorithmus PPO verwendet zur Regulierung der Policy-Updates eine spezielle Verlustfunktion [80]:

$$\mathcal{L}^{\text{CLIP}} = \mathbb{E}_t[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t))]$$

Dabei ist $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{alt}}}(a_t|s_t)}$ das Verhältnis der Wahrscheinlichkeiten der Policy vor dem Update $\pi_\theta(a_t|s_t)$ und nach dem Update $\pi_{\theta_{\text{alt}}}(a_t|s_t)$. A_t bezeichnet die sogenannte Advantage-Funktion, wobei der Advantage definiert ist als $A_t = r + \gamma V(s_{t+1}) - V(s_t)$ und ϵ ist ein frei zu wählender Hyperparameter.

2.2.7 Verteilungsbasiertes Reinforcement Learning

Bei den bisher vorgestellten wertebasierten Verfahren wird versucht eine Wertefunktion zu lernen, um darauf basierend eine Policy ableiten zu können. Der Wert $v(s)$ eines Zustands oder eines Zustands-Aktions-Tupels $q(s, a)$ entspricht dabei dem erwarteten zukünftigen und diskontiertem Erlös. Neben der Betrachtung des Erwartungswertes als skalaren Wert kann es hilfreich sein die

gesamte Werte-Verteilung zu lernen. Durch die Kenntnis der Werte-Verteilung können unterschiedliche Metriken abgeleitet werden beispielsweise zur Modellierung risiko-sensitiver Agenten [58].

Bellemare et al. stellen dazu einen Ansatz vor, bei dem die Werteverteilung durch eine diskrete Verteilung modelliert wird [12]. Grundlage dieser Modellierung ist die sogenannte verteilungsbasierte Bellman-Gleichung:

$$v(s,a) = r(s,a) + \gamma v(s',a')$$

Da es sich bei der Belohnungsfunktion r, dem nächsten Zustands-Aktions-Tupel (s',a') und dem dazu zugehörigen Wert $v(s',a')$ um Zufallsvariablen handelt, wird die Verteilung des Wertes $v(s,a)$ durch diese Zufallsvariablen beeinflusst. $v(s,a)$ wird daher als die Werte-Verteilung bezeichnet [12]. Zum approximativen Erlernen dieser Verteilung mittels Reinforcement Learning schlagen Bellemare et al. den Algorithmus *Approximate Distributional Learning* vor. Dabei wird die Werteverteilung $v(s,a)$ durch eine diskrete Verteilung dargestellt, die durch $N \in \mathbb{N}$ und $V_{\text{MIN}}, V_{\text{MAX}} \in \mathbb{R}$ parametrisiert ist. Die Träger dieser Verteilung sind gegeben durch die Menge $\{z_i = V_{\text{MIN}} + i\triangle z : 0 \leq i < N\}, \triangle z := \frac{V_{\text{MAX}} - V_{\text{MIN}}}{N-1}$. Die atomaren Wahrscheinlichkeiten der Verteilung sind durch ein parametrisches Modell $\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^N$ gegeben, so dass die Wahrscheinlichkeit für den Wert z_i in Abhängigkeit von Zustand s und Aktion a ist [12]:

$$p_i(s,a) := \frac{e^{\theta_i(s,a)}}{\sum_j e^{\theta_j(s,a)}}$$

Durch die Integration der Werte-Verteilung in den Algorithmus *Deep-Q-Networks* [57] entsteht die Architektur *Categorical DQN* [12]. Diese Agenten Modellierung gibt anstelle der Zustands-Aktions-Werte $q(s,a)$ nun jedoch die Wahrscheinlichkeiten $p_i(s,a)$ dafür aus, in einem Zustand s unter der Wahl von Aktion a einen Wert z_i zu beobachten. Basierend auf dieser Wahrscheinlichkeitsverteilung können nun auch andere Entscheidungskriterien zur Aktionswahl eingesetzt werden, deren Entscheidung auf der gesamten Verteilung beruht und nicht nur auf dem Erwartungswert.

2.2.8 Herausforderungen des Reinforcement Learning

Trotz der Erfolge der letzten Jahre gibt es im Reinforcement Learning verschiedene Herausforderungen, die Gegenstand aktiver Forschung in dem Bereich sind. Zu den grundlegendsten Schwierigkeiten gehört der Exploration-Exploitation Trade-Off.

2.2.8.1 Exploration-Exploitation Trade-Off

Alle Reinforcement Learning Algorithmen sehen sich der Schwierigkeit gegenüber, zu steuern, in welchem Maße der Agent zu einem bestimmten Zeitpunkt

neue, möglicherweise unbekannte oder wenig erprobte Aktionen ausprobieren soll (also zu explorieren) oder sich auf bereits erlerntes Wissen zu verlassen, das sich bisher bereits bewährt hat (also bestehendes Wissen auszunutzen). Insbesondere in Umgebungen mit sehr großen Zustandsräumen ist die Exploration eine große Herausforderung, da durch das zufällige Ausprobieren von Aktionen unter Umständen sehr viel Trainingszeit benötigt wird, um sinnvolle Sequenzen von Aktionen zu finden. Für einen Überblick über verschiedene Strategien siehe [52].

2.2.8.2 Sample-Effizienz

Eine weitere Herausforderung liegt in der Minimierung der benötigten Trainingsbeispiele, die ein Agent braucht, um eine gute Policy zu erlernen. Viele Algorithmen benötigen viele Episoden bis gute Policies gelernt werden. Sample-Effizienz ist daher ein wichtiger Schritt, um Reinforcement Learning für Echt-Welt-Probleme anwendbar zu machen.

2.3 Multiagent Reinforcement Learning

Reinforcement Learning ist per Definition ein Einzelagenten-Framework, d.h. es behandelt standardmäßig die Interaktion eines einzelnen Agenten mit einer unbekannten Lernumgebung. Trotz dieser Prämisse wurden Methoden des Reinforcement Learning bereits erfolgreich auf Multiagentensystemen erweitert, was unter dem Namen Multiagent Reinforcement Learning bekannt ist. Die Formale Erweiterung des Markov-Entscheidungsprozesses im Ein-Agenten-Fall ist durch das Stochastische Spiel [47] (auch Markov-Spiel genannt) gegeben. Die Erweiterung durch mehrere Agenten geht mit einer großen Bandbreite unterschiedlicher Modellierungsmöglichkeiten einher. Generell lassen sich zentralisierte und dezentrale Ansätze voneinander abgrenzen. Bei zentralisierten Ansätzen werden Teile der involvierten Komponenten wie die Policy oder die Werte-Funktion zentral gelernt, d.h. es wird ein Modell gelernt, das Informationen aller Agenten aggregiert. Das bringt den Vorteil, dass das Lernproblem stationär bleibt. Andererseits sind zentralisierte Ansätze nicht immer umsetzbar. Das gilt insbesondere dann, wenn davon ausgegangen werden muss, dass die Agenten eigennützig motiviert sind und individuellen Zielen folgen und somit eine Weitergabe von Informationen an eine zentralisierte Instanz nicht gewünscht ist. Im nächsten Absatz soll daher zuerst auf unterschiedliche Modellierungsmöglichkeiten in Multiagentensystemen eingegangen werden, bevor das Stochastische Spiel als Grundlage des Multiagent Reinforcement Learning formal eingeführt wird. Anschließend wird die für diese Arbeit primär eingesetzte Technik des *unabhängigen* Lernens beschrieben und auf die relevanten Metriken dazu eingegangen.

2.3.1 Multiagentensysteme

Ein Multiagentensystem umfasst eine Gruppe autonom handelnder und interagierender Entitäten, die sich eine gemeinsame Umgebung teilen [14, 82, 89, 93]. Durch diese sehr generische Definition lassen sich vielfältige Szenarien als Multiagentensysteme modellieren, wie beispielsweise im Bereich Robotik, im Bereich verteilter Optimierungsprobleme aber auch in sozialwissenschaftlichen Disziplinen wie den Wirtschaftswissenschaften. Generell können Multiagentensysteme danach klassifiziert werden, inwiefern die Ziele der Agenten im Einklang sind bzw. im Widerspruch zueinander stehen (kooperativ, kompetitiv oder gemischt) und auf welcher logischen Ebene Aktionsentscheidungen getroffen werden (zentral oder dezentral). Systeme mit dezentraler Kontrolle, in denen Agenten basierend auf lokalen Zustandsinformationen Entscheidungen treffen, können weiter unterteilt werden, ob Agenten die Möglichkeit besitzen Informationen untereinander auszutauschen also zu kommunizieren oder nicht.

2.3.1.1 Kooperative Systeme

Kooperative Multiagentensysteme sind gekennzeichnet durch eine einheitliche Zielsetzung aller Agenten, d.h. alle Agenten verfolgen das gleiche Ziel. In Bezug auf die Belohnungsfunktion der Agenten bedeutet das, dass alle Agenten die gleichen Belohnungen erhalten. Es gilt also, dass alle Agenten die gleiche Belohnungsfunktion haben: $R^1 = R^2 = ... = R^{\mathcal{N}}$. Kooperative Multiagentensysteme stellen bezüglich der potentiell anwendbaren Lernalgorithmen einige Besonderheiten dar, da in diesem Fall auch die Werte-Funktionen aller Agenten identisch sind und somit auch Methoden aus dem Bereich des Einzelagenten Reinforcement Learning angewendet werden können. Wenn dezentral gelernt wird, also die Agenten individuelle Policies lernen und basierend auf partiellen Beobachtungen Entscheidungen treffen, so kann es zu sogenannten *Credit Assignment Problems* (CAPs) kommen [1, 68]. Dabei liegt die Schwierigkeit darin, dass durch die gemeinsame Belohnungsfunktion einzelne Agenten kein individuelles Feedback zu ihren Aktionen erhalten und somit ihren Beitrag zu der erzielten Gesamtbelohnung nicht zuordnen können.

2.3.1.2 Kompetitive Systeme

In kompetitiven Multiagentensystemen herrscht Wettbewerb zwischen den Agenten. Im Extremfall vollständig kompetitiver Systeme schließen sich die Ziele der Agenten gegenseitig aus. Das wohl bekannteste Szenario umfasst zwei Agenten, wobei die Belohnung des einen Agenten genau dem Verlust des anderen Agenten entspricht, d.h. $R_1 = -R_2$ bzw. $R_1 + R_1 = 0$. Die Summe der Belohnungen ist also immer Null (*zero-sum*). Viele Gesellschaftsspiele wie Schach, Backgammon oder Go fallen in diese Kategorie, da es hier nur jeweils einen Gewinner geben kann. In diesem Fall hängt die Wahl einer optimalen Policy von der Policy des Gegners ab. Beispielsweise wäre im Spiel Schere, Stein,

Papier die Policy immer Stein zu spielen eine optimale Policy gegen einen Gegner, der immer Schere spielt, wohingegen die Policy, die immer Stein spielt gegen die Policy, die immer Papier spielt, verliert. Da die Policy des Gegners normalerweise nicht bekannt ist, lässt sich eine Policy auch danach beurteilen, gegen welchen fiktiven Gegner eine Policy am schlechtesten abschneidet (*worst-case* Kriterium). Werden Policies nach ihrer *worst-case* Performance bewertet, so kann es jedoch sein, dass sehr vorsichtige bzw. konservative Strategien resultieren [47].

2.3.1.3 Gemischte Systeme

In gemischten Multiagentensystemen (engl. *mixed-motive games*) gibt es keine Restriktionen bezüglich der Ziele der Agenten. Grundlegende Annahme ist, dass jeder Agent eine individuelle Belohnungsfunktion besitzt, die die Ziele des Agenten definiert. Im Gegensatz zu (rein) kompetitiven Systemen besteht in gemischten Systemen für die Agenten die Möglichkeit zusammenzuarbeiten bzw. zu kooperieren, da sich ihre Ziele nicht zwangsläufig gegenseitig ausschließen. Kooperation kann sogar dazu führen, dass sich alle Agenten besserstellen verglichen mit dem Fall, dass alle Agenten rein eigennützig handeln. Situationen, in denen Kooperation zwischen eigennützigen Akteuren von essentieller Bedeutung ist, sind Anwendungen wie das autonome Fahren oder Szenarien mit geteilten Ressourcen. Dadurch werden gemischte Systeme zum wichtigsten Anwendungsfall für die Entwicklung von Algorithmen, die darauf abzielen Kooperation zwischen Agenten zu erreichen.

2.3.1.4 Emergenz

In einem Multiagentensystem interagieren mehrere autonome Einheiten in einer geteilten Umgebung. Jeder Agent führt Aktionen gemäß bestimmter Regeln oder gemäß einer (gelernten) Policy aus, wobei Entscheidungen auf Grund individueller und möglicherweise partieller Beobachtungen getroffen werden. Mit wachsender Anzahl von Agenten nimmt die Komplexität zu und eine Vorhersage über die Entwicklung des Gesamtsystems wird zunehmend schwieriger. Dieses sogenannte *emergente* Systemverhalten lässt sich kaum noch vorhersagen, selbst dann nicht, wenn alle individuellen Verhaltensregeln der Agenten bekannt sind, da die Komplexität mit der Anzahl der Individuen sehr schnell wächst. Emergente Phänomene lassen sich beispielsweise bei Tierschwärmen (Vögeln, Fischen und Insekten) beobachten. Reynolds [70] konnte Schwarmverhalten auf drei einfache Regeln reduzieren, womit sich Schwärme simulieren lassen: Kollisionsvermeidung mit Nachbarn, Anpassung der Geschwindigkeit an benachbarte Individuen und Bewegung in Richtung des Zentrums benachbarter Individuen. Folgen alle Individuen diesen Grundregeln so lassen sich Schwarmphänomene erzeugen.

Allgemein ist der Begriff der Emergenz im Kontext Komplexer Adaptiver Systeme (engl. *Complex Adaptive Systems*, (CAS)) geläufig. Ein CAS be-

schreibt ein System bestehend aus vielen einzelnen Komponenten, die miteinander interagieren und die Fähigkeit besitzen sich anzupassen bzw. zu lernen [34]. Dadurch lassen sich viele biologische, chemische oder soziale Systeme modellieren. Da die Komplexität mit der Anzahl der Komponenten stark zunimmt und somit Vorhersagen über die Entwicklung solcher Systeme durch Berechnung nahezu unmöglich werden, ist ein wichtiger Bestandteil der Forschung die computergestützte Simulation. Somit lassen sich emergente Entwicklungen simulieren, um daraus Erkenntnisse abzuleiten. Versteht man beispielsweise eine Gesellschaft als ein komplexes adaptives System, bestehend aus Individuen, die ihr Verhalten und ihre Strategien kontinuierlich an die sich ändernde Welt anpassen, so lassen sich politische oder wirtschaftliche Entwicklungen ebenfalls als emergente Phänomene einordnen [6]. Im Gegensatz zu klassischen Betrachtungen von wirtschaftlichen Entwicklungen, die auf der Berechnung von Gleichgewichten durch die statische Modellierung von Entscheidungsträgern (Firmen, Konsumenten, Investoren) beruhen, lässt sich dadurch eine sich ständig verändernde und dynamische Gesellschaft modellieren.

2.3.2 Stochastisches Spiel

Formal wird ein Multiagentensystem als Stochastisches Spiel [47] definiert und stellt die natürliche Erweiterung eines Markov-Entscheidungs-Prozesses für Systeme mit mehr als einem Agenten dar. Das stochastische Spiel (auch Markov-Spiel genannt) ist formal definiert als Tupel $\mathcal{G} = (\mathcal{N}, \mathcal{A}^1, ..., \mathcal{A}^{\mathcal{N}}, \mathcal{S}, \mathcal{T}, \mathcal{R}^1, ..., \mathcal{R}^{\mathcal{N}})$, das sich aus folgenden Komponenten zusammensetzt: zunächst gibt es eine endliche Menge an Agenten (Spielern) $\mathcal{N}$. Jeder Agent $i \in N$ verfügt über eine Menge an Aktionen $\mathcal{A}^i$, die ausgeführt werden können. Es gibt eine Menge von Zuständen $\mathcal{S}$, wobei durch die Ausführung der Aktionen aller Agenten ein neuer Zustand hervorgerufen wird. Der Übergang von einem Zustand in einen neuen Zustand wird durch die Zustandstransitionsfunktion $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \rightarrow PD(\mathcal{S})$ gesteuert, wobei $PD(\mathcal{S})$ die Menge an diskreten Wahrscheinlichkeitsverteilungen über $\mathcal{S}$ beschreibt. Die Transitionen sind also nicht notwendigerweise deterministisch, sondern können auch gemäß einer Wahrscheinlichkeitsverteilung erfolgen. Befindet sich die Umgebung zum Zeitpunkt t also in einem Zustand s_t, so erfolgt der Übergang in einen neuen Zustand s_{t+1} basierend auf den Aktionen $\mathbf{a}_t$. Schlussendlich hat jeder Agent eine Belohnungsfunktion $\mathcal{R}^i : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \rightarrow \mathbb{R}$, die das eigentliche Ziel des Agenten definiert. Beim Übergang von Zustand s_t auf Zustand s_{t+1} basierend auf den Aktionen $\mathbf{a}_t$ erhält Agent i eine numerische Belohnung $\mathcal{R}_i(s_t, a_t, s_{t+1})$.

2.3.3 Unabhängiges Lernen

Der in dieser Arbeit primär verwendete Ansatz zur Modellierung des Lernprozesses wird als unabhängiges Lernen (engl. *independent learning*) bezeichnet. Dabei wird davon ausgegangen, dass jeder Agent $i \in \mathcal{N}$ eine eigene Instanz

eines Lernalgorithmus darstellt. Jeder Agent lernt somit eine individuelle Policy π basierend auf seinen eigenen lokalen Beobachtungen und versucht die eigenen zu erwartenden Belohnungen zu maximieren. Dieser Ansatz bringt den Vorteil algorithmischer Einfachheit und lässt sich durch natürliche adaptive Multiagentensysteme motivieren: Der Mensch lernt und trifft Entscheidungen basierend auf individuellen Erfahrungen. Gerade für die Untersuchung von entstehender Kooperation ist dieser Ansatz naheliegend, da somit gewährleistet ist, dass jedes Individuum *a priori* eigennützig motiviert ist und sich dennoch zu kooperativem Handeln bewegen lässt. Aus Sicht des zugrundeliegenden Lernproblems erzeugt dieser Ansatz jedoch die Schwierigkeit der Nicht-Stationarität: Im Gegensatz zum Einzel-Agenten Reinforcement Learning verändert sich aus Sicht eines einzelnen Agenten eines lernenden Multiagentensystems das Lernproblem ständig, denn durch das parallele und unabhängige Lernen mehrerer Agenten, ist eine optimale Policy zu einem späteren Zeitpunkt unter Umständen nicht mehr optimal, da die Optimalität immer von den aktuellen Policies aller Agenten abhängig ist [44]. Da alle Agenten ihre Policies kontinuierlich verändern, entsteht aus Sicht des einzelnen Agenten ein *Moving Target*-Lernproblem, d.h. die eigene Policy muss kontinuierlich an die Policies aller Agenten angepasst werden. Durch Nicht-Stationarität können Konvergenz-Eigenschaften mancher Algorithmen verletzt werden [44]. In der Praxis lassen sich jedoch trotz dieser Limitierung viele Probleme erfolgreich durch den Ansatz unabhängiger Agenten erlernen [45, 95].

2.3.4 Metriken

Im Einzel-Agenten Reinforcement Learning ist das Ziel, eine optimale Policy zu lernen, d.h. eine Policy, die den erwarteten, langfristigen, diskontierten Erlös maximiert. Um die Qualität einer Policy zu bewerten, wird meistens die episodisch kumulierte Summe der Belohnungen verwendet. Über den Verlauf der Trainingsepisoden lässt sich somit darstellen, wie schnell der Agent lernt und ob sich das Training stabil entwickelt bzw. ob es hohe Varianz während des Trainingsverlaufs gibt. Für den Multiagenten Fall in gemischten Systemen gibt es keine einfache skalare Metrik, die den Lernerfolg so abbildet wie im Einzel-Agenten-Fall. Relevant für die Bewertung der Entwicklung eines Systems unabhängig lernender Agenten ist das Gesamtergebnis (engl. *social outcome*), das durch verschiedene Metriken abgebildet werden kann. Jeder Agent i erhält seine individuellen Belohnungen $\{r_t^i | t = 1, ...T\}$ für eine Episode der Länge T. Der episodische Erlös eines Agenten ist somit $\mathcal{R}^i = \sum_{t=1}^{T} r_t^i$. In dieser Arbeit werden primär zwei Metriken zur Bewertung verschiedener Verfahren verwendet:

- Die Utilitaristische Metrik (U) [66] (auch Gesamtbelohnung oder Effizi-

enz genannt) misst die Summe aller Belohnungen aller Agenten:

$$U = \sum_{i=1}^{|\mathcal{N}|} \mathcal{R}_i$$

Diese Metrik bildet somit die Gesamtentwicklung des Systems ab, es werden jedoch keine individuellen Entwicklungen berücksichtigt.

- Eine Metrik, mit der sich verschiedene Zustände auf Individualebene vergleichen lassen, ist das Prinzip der *Pareto-Effizienz*. Ein Zustand ist Pareto-Effizient, wenn es keinen Zustand gibt, bei dem mindestens ein Agent besser gestellt werden kann, ohne gleichzeitig einen anderen Agenten schlechter zu stellen. Die Menge aller Pareto-Effizienten Zustände wird als Pareto-Front bezeichnet.

3 Marktmechanismen in stochastischen Spielen

Ziel dieser Arbeit ist es, marktbasierte Methoden vorzustellen, die dazu beitragen Effizienzprobleme zwischen eigennützigen Agenten aufzulösen bzw. abzumildern. Dazu werden in diesem Kapitel zwei Aspekte behandelt: Der erste Abschnitt stellt dar, wie sich ein gegebenes Multiagentensystem so erweitern lässt, dass die Agenten als Handelspartner aktiv werden können. Das formale Konzept dazu wird in Anlehnung an das stochastische Spiel [47] als *stochastisches Marktspiel* bezeichnet. Der freie Fluss von Ressourcen zwischen den Akteuren, soll dabei helfen, ineffiziente Allokationen von Ressourcen auf die Agenten zu verhindern. Durch den Austausch zwischen Agenten werden einzelnen Agenten Anreize zu kooperativem Handeln gegeben. Unterschieden werden dabei *konditionale* und *bedingungslose* Ansätze, die sich dahingehend unterscheiden, ob Transfers an Bedingungen gebunden sind oder nicht. Die Höhe des realisierten Transfers wird dabei nicht von den Agenten verhandelt, sondern wird exogen vorgegeben. Die Agenten entscheiden also nicht wie viel sie handeln mögen, sondern ob sie handeln mögen. Die Höhe des Gesamttransfers kann lediglich über die Menge an Transaktionen gesteuert werden.

Der zweite Abschnitt behandelt daher die Frage, wie sich ein Verhandlungsprozess zwischen zwei eigennützigen Agenten, also beispielsweise eine Preisbildung modellieren lässt und von welchen Faktoren das Verhandlungsergebnis beeinflusst wird. Dabei werden zwei unterschiedliche Szenarien untersucht, wobei das erste eine Aufteilungsverhandlung und das zweite eine Verkaufssituation zwischen einem Käufer und einem Verkäufer darstellt. Zu den untersuchten Agenten-internen Einflussfaktoren gehört die Risikoaversion, die über eine spezielle Variante eines Reinforcement Learning-Algorithmus gesteuert werden kann. Zu den untersuchten externen Einflussgrößen gehört die Informationsasymmetrie zwischen den Verhandlungspartnern.

3.1 Vorveröffentlichungen

Die Kerninhalte dieses Kapitels wurden vom Autor in [76] bzw. in [77] publiziert. Die Definition des stochastischen Marktspiels sowie die empirische Evaluation dazu wurden vom Autor in [76] veröffentlicht. Wie in Kapitel 1.1 dargestellt, stammen die in den Vorveröffentlichungen und im Folgenden präsentierten Inhalte bezüglich der Idee, des Konzepts und der Evaluation vom Autor der vorliegenden Arbeit. Die in diesem Kapitel dargestellten Abbildungen sind

der Art nach ebenfalls bereits in [76] bzw. in [77] veröffentlicht, wurden im Rahmen der Ausarbeitung dieser Arbeit aber neu visualisiert und beschriftet. Die Datengrundlage für die Abbildungen hat sich dabei jedoch nicht geändert und entspricht den gleichen Auswertungen wie in [76] bzw. in [77] vorgestellt.

3.2 Stochastische Marktspiele

In Multiagentensystemen mit gemischten Motiven (siehe Abschnitt 2.3.1), in denen die Agenten unterschiedliche Ziele verfolgen, besteht ein zentrales Hindernis für die kollektive Zusammenarbeit darin, dass die Entscheidungsträger nicht wissen, wie sich ihr eigenes Verhalten auf das Ergebnis anderer auswirkt. Diese Herausforderung bzgl. einer möglichen Zusammenarbeit zwischen den Agenten wird als fehlendes Verständnis über die Ziele und Motive anderer Agenten aufgeführt (engl. *Understanding*) [21]. Um zu kooperativem Handeln befähigt zu sein, ist es demnach wichtig ein Verständnis über die Erwartungen, Ziele, Anreize und Fähigkeiten anderer Agenten zu haben. Solange einzelne Agenten keinerlei Informationen darüber erhalten, wie sich ihre Handlungen auf die Zielerreichung anderer Agenten auswirken, noch über die Kosten, die sie bei anderen verursachen können, ist demnach kein Anreiz dafür gegeben, das eigene Verhalten zu verändern. Die Auswirkungen des eigenen Handelns auf andere Akteure sind daher aus Sicht des Individuums *extern*, da sie in den individuellen Belohnungsfunktionen nicht berücksichtigt werden. Darüber hinaus gibt es für die einzelnen Akteure keinen Anreiz diese externen Kosten zu internalisieren, solange sie dafür nicht entschädigt werden oder eine Gegenleistung erhalten. Das kann eine sinnvolle Zusammenarbeit verhindern, da unter solchen Umständen die rationale Entscheidungsfindung des Einzelnen nicht notwendigerweise zu einem optimalen Gesamtergebnis im Sinne aller Agenten führt.

Die Idee zu dem in diesem Abschnitt vorgestellten Konzept liegt darin die Kräfte des Marktes zu nutzen, um in lernenden Multiagentensystemen Anreize zu schaffen, die zu einer verbesserten Koordination und Kooperation zwischen den Agenten führen. Die Idee motiviert sich aus dem Einsatz von Marktmechanismen in menschlichen Gesellschaften, von denen bekannt ist, dass Märkte einen sehr effektiven Mechanismus darstellen, um potentiell große Mengen eigennütziger Einheiten dezentral zu organisieren. Auch aus theoretischer Sicht sind Märkte gut erforscht, wobei die Literatur zurückreicht bis hin zu Adam Smith und der unsichtbaren Hand des Marktes [84]. Märkte bieten verschiedene Attribute, die sie für den Einsatz im Bereich Multiagentensysteme interessant machen, dazu gehören Offenheit und Skalierbarkeit. Offenheit ist insbesondere für zukünftige KI-Systeme relevant, die sich aus unterschiedlichen Entitäten zusammensetzen können, darunter u.U. regelbasierte Verfahren, kontinuierlich lernende KIs sowie menschliche Akteure. Skalierbarkeit ist ein entscheidender Faktor, wenn ein Einsatz in echten Anwendungen realisiert werden soll. Märkte sind darüber hinaus besonders effektiv bei der Förderung einer effizienten,

kooperativen Interaktion zwischen heterogenen Teilnehmern, weil (idealisierte) Märkte besondere Regeln (Eigentumsrechte, fälschungssichere Währungen, Warenzeichen) definieren, so dass kooperatives Verhalten zu einer dominanten Strategie im spieltheoretischen Sinne wird [55].

3.2.1 Verwandte Arbeiten

Erste Konzepte und Frameworks zur Nutzung von Marktmechanismen in verteilten Systemen gehen zurück auf Miller und Drexler [54, 55], motiviert durch die Beobachtung, dass durch die zunehmende Dezentralisierung komplexer Rechenaufgaben auch der Aufwand steigt einzelne Prozesse zu koordinieren. Die Idee, Koordination basierend auf Handels- und Preis-Mechanismen umzusetzen, wurde dabei in dem Framework *agoric systems* beschrieben [55]. Marktbasierte Ansätze wurden im Kontext von lernenden Systemen bisher insbesondere in rein kooperativen Multiagentensystemen untersucht [18, 33, 55]. Im Gegensatz dazu werden Märkte in dieser Arbeit für den Einsatz in gemischten Systemen, d.h. in Umgebungen, in denen die Agenten eigennützig handeln und keine kooperativen Eigenschaften besitzen verwendet.

3.2.1.0.1 Märkte in rein kooperativen Multiagentensystemen Der wohl erste marktbasierte Algorithmus im Kontext von KI-Systemen wurde von Holland entwickelt und ist bis heute unter dem Namen *Bucket Brigade* bekannt [33]. Der Algorithmus arbeitet auf der Grundlage von Entitäten, sogenannten Klassifikatoren, die jeweils aus einer Bedingung und einer Aktion bestehen. Wenn der Systemzustand mit der Bedingung einer Entität übereinstimmt, wird ein Gebot berechnet, und die Entität mit dem höchsten Gebot darf ihre spezifische Aktion in eine Ausführungswarteschlange stellen. Das Gebot wird auf die zuvor aktiven Klassifikatoren verteilt, die das System zur auslösenden Bedingung geführt haben, so dass Belohnungsketten entstehen und ein Lernprozess stattfinden kann. Basierend auf diesem Konzept wurde die *Hayek machine* entwickelt [10], die den Algorithmus Bucket Brigade um Eigentumsrechte erweitert. Die Entitäten (Agenten) können nun in ihre eigenen Nachkommen investieren, um später von deren Gewinnen zu profitieren. Einen Ansatz für marktbasiertes Reinforcement Learning (RL) zum Lernen in teilweise beobachtbaren Markov-Entscheidungsprozessen (engl. Partially Observable Markov Decision Processes, (POMDPs)) wurde in [43] vorgeschlagen, wobei der Ansatz auf einer Variante der Hayek-Maschine mit einem zusätzlichen Speicherregister basiert, das von Agenten mit spezifischen Schreibaktionen manipuliert werden kann. In jüngerer Zeit wurde die Idee des Kaufs und Verkaufs des Rechts, in der Welt zu handeln, durch einzelne Agenten in [18] aufgegriffen und mit verschiedenen RL-Algorithmen kombiniert, um einen allgemeinen Rahmen für die Organisation einer Gesellschaft von Agenten zu formalisieren.

Die in dieser Arbeit vorgestellten Ansätze unterscheiden sich von den oben beschriebenen Konzepten dadurch, dass in dieser Arbeit ausschließlich ge-

mischte Multiagentensystemen betrachtet werden, in denen die Agenten individuelle Ziele verfolgen. Der Marktmechanismus dient dabei als Katalysator für die Zuweisung von Ressourcen an Agenten, um somit die Gesamteffizienz zu verbessern, zielt also nicht lediglich darauf ab potentielle Koordinationsprobleme zwischen kooperativen Agenten zu lösen.

3.2.1.0.2 Kooperation in gemischten Multiagentensystemen In den letzten Jahren wurden einige Arbeiten vorgestellt, die sich mit der Frage von Kooperation zwischen eigennützig motivierten KI-Agenten beschäftigen. So wird in [66] Multi-Agent Reinforcement Learning zur Untersuchung spieltheoretischer Gleichgewichte verwendet, in einem Szenario, in dem sich die Agenten eine gemeinsame Pool-Ressource teilen (engl. *Common Pool Ressource*, (CPR)). Dabei werden auch räumliche Aspekte der Domäne berücksichtigt, es wird also untersucht, inwiefern die räumliche Anordnung bzw. Verteilung der Agenten innerhalb der Domäne das Resultat beeinträchtigt. Das innovative dieses Ansatzes liegt darin, dass anstelle von spieltheoretischen Analysen Strategien durch RL-Agenten in komplexen Domänen erlernt werden, um somit die daraus resultierenden emergenten Entwicklungen zu analysieren. Dadurch können verschiedene Einflüsse durch Computer-Simulationen analysiert werden, anstatt rigorose mathematische Modelle zu verwenden, die aufgrund der hohen Komplexität kaum berechenbar sind. Ein ähnlicher Ansatz wird auch von Leibo et al. in [45] verwendet, wobei die Autoren sich insbesondere mit der Frage beschäftigen, wie sich soziale Dilemmas näher an Echt-Welt-Situationen heranführen lassen. Dafür schlagen sie das sequentielle soziale Dilemma (SSD) vor, das verschiedene Aspekte wie die zeitliche Erweiterung der Agenten-Interaktion und feingranulare Abstufungen von Kooperation zwischen den Agenten umfasst. In verschiedenen sequentiellen sozialen Dilemmas zeigen die Autoren, das exogene Einflüsse der Umgebung, wie beispielsweise die Menge an vorhandenen Ressourcen, den entstehenden Kooperationsgrad zwischen den Agenten maßgeblich beeinflussen können.

SSD-Szenarien werden seither verwendet, um kooperative Algorithmen zu testen, so wie von Lupu und Precup [49], deren Mechanismus auf Schenkungen (engl. gifting) zwischen den Agenten beruht, wodurch es den Agenten möglich ist sich direkt Belohnungen zuzuweisen. Ein weiterer Ansatz, kooperatives Verhalten in SSD-Umgebungen mittels Reinforcement Learning zu erzeugen, beruht auf der Abbildung bekannter erfolgreicher spieltheoretischer Strategien. Eine Strategie, die sich dabei als äußerst stabil herausgestellt hat, ist *Tit-for-Tat* [7], einer Strategie, die darauf beruht im ersten Zug kooperativ zu sein und anschließend genau die Aktionen des Gegenübers aus dem letzten Schritt zu kopieren. Ein Ansatz, Tit-for-Tat in komplexe Umgebungen zu übertragen, wurde von Lerer und Peysakhovich [46] gemacht. In ihrer Arbeit konstruieren sie Tit-for-Tat-Agenten und zeigen durch Experimente und theoretisch ihre Fähigkeit, Kooperation aufrechtzuerhalten, während rein reaktive Trainingstechniken eher zu sozial ineffizienten Ergebnissen führen.

Diese Arbeit folgt dem Ansatz der oben vorgestellten Arbeiten, indem Multi-Agent Reinforcement Learning eingesetzt wird, um einen Lernprozess mehrerer unabhängiger Agenten zu modellieren. Dabei werden die emergenten Ergebnisse von Agenten mit und ohne Märkte verglichen. Es wird jedoch nicht versucht bestimmte bewährte Strategien zu erlernen, um somit Kooperation zu erhöhen wie in [46]. Die in dieser Arbeit betrachteten Evaluationsumgebungen sind dabei stark von dem Konzept der sequentiellen sozialen Dilemmas inspiriert (vgl. Smartfactory in Abschnitt 4.4.1). Der Gifting-Mechanismus ähnelt in seiner Funktionsweise einem Marktansatz, da er den Aktionsraum um bestimmte Gifting-Aktionen erweitert. Er unterscheidet sich jedoch von der Idee eines Marktes, da es keinen tatsächlichen Handel zwischen Agenten auf der Grundlage eines Kauf- und Verkaufsangebots gibt.

3.2.2 Konzept

Im Folgenden wird nun der Ansatz zur Integration eines Marktmechanismus in ein Multiagentensystem vorgestellt. Dabei wird auf der allgemeinen Definition des stochastischen Spiels aufgebaut (vgl. Abschnitt 2.3.2), wobei die Erweiterung als *stochastisches Marktspiel* bezeichnet wird. Sei $\mathcal{G} = (\mathcal{N}, \mathcal{A}^1, ..., \mathcal{A}^{\mathcal{N}}, \mathcal{S}, \mathcal{T}, \mathcal{R}^1, ..., \mathcal{R}^{\mathcal{N}})$ ein stochastisches Spiel, dann wird $\mathcal{G}$ durch die Hinzunahme folgender Komponenten in ein stochastisches Marktspiel überführt:

- Zusätzlich zu den originalen Aktionsräumen $\mathcal{A}^i$ für $i \in \mathcal{N}$, erhält jeder Agent $i \in \mathcal{N}$ eine Menge von Marktaktionen $\mathcal{A}^i_m$. Der Aktionsraum für i des stochastischen Marktspiels $\mathcal{A}^i_{\text{SMS}}$ ergibt sich dann aus dem kartesischen Produkt des Aktionsraums $\mathcal{A}^i$ und des Marktaktionsraums $\mathcal{A}^i_m$, also $\mathcal{A}^i_{\text{SMS}} = \mathcal{A}^i \times \mathcal{A}^i_m$.
- Eine Marktfunktion $\mathcal{M} : \mathcal{S} \times \mathcal{A}^1_{\text{SMS}} \times ... \times \mathcal{A}^N_{\text{SMS}} \times \mathbb{R}^N \times \mathbb{R}^N$, die die Umverteilung der Belohnungen, basierend auf dem Zustand $s \in \mathcal{S}$, den Aktionen der Agenten und der aktuellen Belohnungen berechnet.
- Einem Preis p, der die Höhe eines Transfers zwischen zwei Agenten definiert.

Durch den erweiterten Aktionsraum $\mathcal{A}^i_{\text{SMS}}$ kann Agent i als Marktteilnehmer auf den Markt Einfluss nehmen. Das bedeutet, dass Agenten in dem erweiterten Spiel zusätzlich zu ihren originalen Aktionen $a_e \in \mathcal{A}^i$ auch Marktaktionen $a_m \in \mathcal{A}^i_m$ wählen müssen. Abbildung 3.1 verdeutlicht den Ablauf für den Fall zweier Agenten. Die Marktfunktion ist zuständig für die Umverteilung der Belohnungen, basierend auf den Marktaktionen der Agenten. Marktfunktionen können dabei verschiedene Formen annehmen, je nachdem nach welchem Prinzip der Markt arbeiten soll. Im Rahmen dieser Arbeit werden zwei Arten von Marktfunktionen $\mathcal{M}$ unterschieden, die als **bedingte Märkte** und **unbedingte Märkte** bezeichnet werden. Ein unbedingter Markt erlaubt den

Handel zwischen zwei Agenten, der nicht durch weitere Bedingungen eingeschränkt ist. Das heißt der Austausch von Belohnungen ist an keine bestimmten Bedingungen geknüpft. Wenn Agenten über einen unbedingten Markt Belohnungen austauschen, müssen sie sich daher nicht auf ein bestimmtes Verhalten, bestimmte Regeln oder Normen gegenüber dem Handelspartner verpflichten. Der unbedingte Handel stellt somit eine Möglichkeit dar, die Ziele der Agenten miteinander zu verbinden, ohne das individuelle Verhalten der Agenten zu reglementieren. Im Gegensatz dazu definiert ein bedingter Markt einen Handel als einen Austausch, der an bestimmte weitere Bedingungen geknüpft ist. Die Agenten können somit den Austausch einer Belohnung davon abhängig machen, ob der Handelspartner bestimmte Aktionen ausgeführt hat oder ob bestimmte Systemzustände eingetreten sind. Im Folgenden werden die beiden Ansätze näher beschrieben.

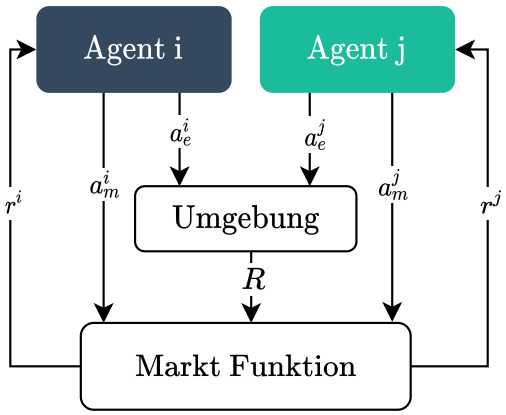


Abbildung 3.1: Im stochastischen Marktspiel wählen die Agenten sowohl Aktionen aus dem originalen Aktionsraum als auch Marktaktionen, um mit anderen Agenten zu Handeln.

3.2.3 Bedingungslose Märkte

Bedingungslose Märkte beruhen auf der Idee, dass es den Agenten eines Multiagentensystems ermöglicht werden muss, andere Akteure an ihrer eigenen Belohnung teilhaben zu lassen, um ihre Ziele besser in Einklang zu bringen. Vergleichen lässt sich dieses Prinzip mit einer Aktiengesellschaft: Durch den Erwerb von Unternehmensanteilen wird der Aktionär an zukünftigen Unternehmensgewinnen beteiligt. Die Auszahlung an die Aktionäre ist dabei nicht an bestimmte Bedingungen bezüglich des Verhaltens der Aktionäre gebunden, d.h. die Aktionäre müssen sich ihren Anteil nicht über bestimmte Leistungen verdienen. Dieses Prinzip, das im englischen als *Shareholding* bekannt ist, ist die zugrundeliegende Idee der bedingungslosen Märkte dieser Arbeit. Eine bedingungslose Marktfunktion ermöglicht es daher jedem Agenten, als Emittent von Aktien aufzutreten, d.h. er kann Anteile seiner eigenen zu erwartenden

Belohnung an andere Agenten ausgeben. Gleichzeitig kann ein Agent Anteile von anderen Agenten beziehen, wodurch ihm eine Beteiligung ermöglicht wird. Durch die in dieser Art und Weise geschaffene Vernetzung der Agenten werden auch implizit deren Ziele aneinander angeglichen. Die Idee ist, dass ein Agent, der Anteile eines anderen Agenten hält, weniger Anreize hat diesen Agenten an der Erreichung seiner Ziele zu hindern, da sich dadurch auch seine eigene erwartete Belohnung schmälern würde. Somit sollten im Idealfall die Agenten, die die höchste zu erwartende Belohnung in Aussicht haben von anderen Agenten dabei unterstützt (oder zumindest nicht daran gehindert) werden, ihre Ziele zu erreichen, wodurch alle Agenten potentiell profitieren können.

Die Agenten in einem stochastischen Marktspiel mit bedingungsloser Marktfunktion, entscheiden also zu jedem Zeitschritt, ob sie Anteile von anderen Agenten kaufen und ob sie ihre eigenen Anteile an andere Agenten veräußern wollen. Durch den direkten Bezug des bedingungslosen Marktes zu einem Aktienmarkt, werden die Begriffe bedingungsloser Markt und Aktienmarkt in dieser Arbeit synonym verwendet. Im bedingungslosen Markt gibt die Marktaktion a_m^i des Agenten i die Bereitschaft an, von welchen Agenten Anteile gekauft werden sollen bzw. an welche Agenten eigene Anteile ausgegeben werden sollen. Wenn ein Kaufangebot des Agenten i an j mit einem Verkaufsangebot von j an i zusammentrifft, so wird eine Transaktion zwischen i und j initiiert. Durch den dadurch erworbenen Anteil wird Agent i zukünftig mit einem bestimmten Anteil d (der Dividende) an den zukünftigen Belohnungen von j beteiligt. Optional können Aktien auch einen Preis p haben, der für den Kauf der Aktie bezahlt werden muss. Ein solcher Preis ist jedoch nicht zwingend erforderlich, da Agenten im Zweifelsfall bereit sein werden ihre Anteile kostenlos abzugeben, um andere Agenten zu Anteilseignern zu machen, um ihre Ziele anzugleichen.

3.2.4 Konditionale Märkte

Konditionale Märkte beruhen auf der Idee, einem Agenten i zu erlauben, eine Belohnung an Agent j zu bezahlen, basierend auf dem (vorausgehenden) Verhalten von j. In diesem Sinne definiert ein konditionaler Markt einen Kaufvertrag zwischen zwei Parteien, einer Verkäufer- und einer Käuferseite, ähnlich wie in der realen Welt, wo sich zwei Personen auf eine Auszahlung einigen, wenn bestimmte (Vertrags)-Bedingungen erfüllt sind. Hier betrachten wir den Fall, in dem diese Bedingungen über die Marktaktionen $\mathcal{A}_m^i$ spezifiziert werden können und entweder auf dem aktuellen Zustand s_t oder auf den ausgeführten Aktionen $a^i \in \mathcal{A}_e^i$, die von Agenten ausgeführt werden, basieren können. Ein Beispiel für einen bedingten Markt, bei dem die Bedingungen auf den Handlungen der Agenten basieren, wurde in [74] unter dem Begriff Aktionsmarkt (AM) vorgestellt. In einem Aktionsmarkt umfasst die Aktion a_t^i des Agenten i zum Zeitpunkt t sowohl eine eigene Aktion (Umweltaktion) $a_t^{i,\text{env}}$ als auch eine mögliche Angebotsaktion $a_t^{i,\text{offer},j}$, die dem Agenten j zum Handel vorgeschla-

gen werden kann. Wenn j die vorgeschlagene Aktion zum Zeitpunkt t ausführt, d.h. wenn $a_t^{j,\text{env}}$ gleich $a_t^{i,\text{offer},j}$ ist, dann kommt ein Handel zwischen i und j zustande. In diesem Fall ist i verpflichtet, j den festgelegten Preis p in Form von Belohnungseinheiten zu zahlen.

3.2.5 Training unabhängiger Agenten mit Markt

In diesem Abschnitt wird nun der Trainingsprozess für das Training $\mathcal{N}$ unabhängiger Agenten in einem stochastischen Marktspiel skizziert. Dabei handelt es sich um einen episodenbasierten Trainingsansatz, das Training gliedert sich also in Episoden mit fester Länge. Jede Episode beginnt mit einer zufälligen Anfangskonfiguration der Umgebung und das Training erfolgt über den Verlauf einer festen Menge von Episoden E. Der Ablauf der Trainingsprozedur ist als Pseudocode in Algorithmus 1 dargestellt. Zu jedem Zeitschritt t einer Episode e wählt Agent i eine Aktion a^i gemäß seiner aktuellen Policy $\pi(s_t|\theta^i)$ parametrisiert durch die Gewichte θ^i basierend auf Zustand s_t. Basierend auf dem Vektor aller Aktionen $\mathbf{a}_t$ geht die Umgebung durch die Zustandstransitionsfunktion $\mathcal{T}$ in einen Folgezustand s_{t+1} über, und die Belohnungen $\mathbf{r}_t$ der Agenten werden über die Belohnungsfunktion $\mathcal{R}$ ermittelt. Anschließend erfolgt der Aufruf der Marktfunktion $\mathcal{M}$, die, basierend auf den Aktionen der Agenten $\mathbf{a}_t$, den Belohnungen $\mathbf{r}_t$ und dem aktuellen Zustand s_t, eine Umverteilung der Belohnungen berechnet. Das Ergebnis der Umverteilung hängt von der konkreten Instanz des Marktes ab (bedingt oder bedingungslos). Es folgt ein Trainingsschritt basierend auf den Erfahrungen der Agenten. Im Falle eines Trainingsverfahrens mit Replay Memory wird dazu eine Charge der Größe C an gesammelten Erfahrungen bestehend aus Zuständen s, Aktionen a^i, Folgezuständen s' und Belohnungen r^i aus dem Erfahrungsspeicher des Agenten gezogen und für einen stochastischen Gradienten-Schritt verwendet.

Algorithm 1 Trainingsprozedur unabhängiger Agenten mit Markt

1: **for** episode $e \in E$ **do**
2: **for** step $t \in T$ **do**
3: **for** agent $i \in \mathcal{N}$ **do**
4: $a_t^i \sim \pi^i(s_t|\theta^i)$ ▷ Aktionswahl
5: $s_{t+1} \leftarrow \mathcal{T}(\mathbf{a}_t|s_t)$ ▷ Episoden-Schritt
6: $\mathbf{r}_t \leftarrow R(\mathbf{a}_t|s_t)$ ▷ Belohnungen
7: $\mathbf{r}_t \leftarrow \mathcal{M}(\mathbf{a_t}, \mathbf{r}_t, s_t)$ ▷ Aufruf der Marktfunktion
8: **for** agent $i \in \mathcal{N}$ **do**
9: trainiere θ^i basierend auf $\{(s, a^i, s', r^i)_c : c = 1, .., C\}$

Die Funktionsweise der Marktfunktion ist in Algorithmus 2 dargestellt. Die Marktfunktion erhält dazu den aktuellen Zustand s_t, die Aktionen $\mathbf{a}_t$ sowie die Belohnungen $\mathbf{r}_t$. Je nach Art des verwendeten Marktes kann es notwendig sein, dass eine Bilanz geführt wird. In der Bilanz werden die Verbindlichkeiten

bzw. Forderungen der Agenten gegenüber anderen Agenten gespeichert. Verbindlichkeiten des Agenten i gegenüber Agent j entstehen, wenn i noch eine offene Zahlungsverpflichtung gegenüber j hat, die durch eine vorangegangene Transaktion zwischen i und j vereinbart wurde. Die Bedingungen, wann eine solche Verbindlichkeit beglichen wird, hängen von der konkreten Marktform und den Möglichkeiten der Agenten zum Schulden machen ab. Da die Verbindlichkeiten und Forderungen Teil des Zustands s_t der Umgebung sind, lässt sich die Bilanz b_t über s_t ermitteln.

Zur Berechnung der Umverteilung der Belohnungen gemäß der Marktfunktion $\mathcal{M}$ werden zwei Matrizen $S, B \in 0_{N,N}$ erzeugt. Anschließend wird über alle Agenten $i \in \mathcal{N}$ iteriert, und für jeden Agenten i werden die Verkaufs- und Kaufangebote für die anderen Agenten $j \in \mathcal{N} \setminus i$ in den Matrizen S und B vermerkt. Zur Speicherung bietet sich ein *One-Hot-Encoding* an, bei dem für jedes erfolgte Angebot von Agent i an Agent j eine 1 in Zelle (i, j) eingetragen wird. Nachdem alle Angebote ermittelt wurden, werden die Transaktionen T zum Zeitschritt t über die logische Operation $\wedge$ (Verundung) auf den Matrizen S und B ermittelt. Die Berechnung der neuen Belohnungen erfolgt dann gemäß der aktuellen Transaktionen T, den Forderungen und Verbindlichkeiten aus der Bilanz b_t, sowie der aktuellen Belohnungen $\mathbf{r}_t$. Weitere Umverteilungsaspekte hängen dann von der konkreten Spezifikation der Marktfunktion ab.

Algorithm 2 Marktfunktion

procedure $\mathcal{M}(s_t, \mathbf{a}_t, \mathbf{r}_t)$
  $b_t \leftarrow s_t$ ▷ Bilanz
  $S, B \leftarrow 0_{N,N}$ ▷ Verkauf-(S) und Kaufangebote (B)
  for agent i in $\mathcal{N}$ **do**
    for agent j in $\mathcal{N} \setminus i$ **do**
      $S_{i,j} \leftarrow a_t^{i,\text{sell},j}$ ▷ Verkaufsangebote von i an j
      $B_{i,j} \leftarrow a_t^{i,\text{buy},j}$ ▷ Kaufangebote von i an j
  $T \leftarrow S \wedge B$ ▷ Berechnung der Transaktionen
  set rewards $\mathbf{r}_t$ w.r.t. $T, b_t, \mathbf{r_t}$
  return $\mathbf{r}_t$

Aus Software-technischer Sicht, lässt sich die Umgebung und die Marktfunktion lose koppeln, d.h. die Marktfunktion wird in Form einer eigenen Klasse bereitgestellt und zur Laufzeit als Attribut an die Umgebung gebunden. Die lose Kopplung der beiden Komponenten bietet sich hier insbesondere an, da somit nur geringfügige Änderungen an der Umgebung vorgenommen werden müssen, was erforderlich ist, da die Umgebungen oft aus bestehenden Simulations- und Testumgebungen eingebunden werden.

	a_1	a_2
a_1	$1, 0$	$0, 1$
a_2	$0, 1$	$1, 0$

(a) Konflikt

	a_1	a_2
a_1	$1, 1$	$0, 0$
a_2	$0, 0$	$1, 1$

(b) Koordination

Abbildung 3.2: Matrix-Spiel mit Konflikt- und Koordinations-Problem zwischen den Agenten.

3.2.6 Einsatzmöglichkeiten eines Marktes

Im Folgenden sollen die potentiellen Einsatzmöglichkeiten des Marktkonzepts für Multiagentensysteme analysiert werden. Wie in Abschnitt 2.3.1 beschrieben, können die Beziehungen zwischen zwei Agenten rein *kompetitiv*, *gemischt* oder rein *kooperativ* sein. Für rein kompetitive Situationen, die durch die Nullsummen-Eigenschaft geprägt sind, gilt, dass Kooperationen zwischen den Agenten ausgeschlossen sind (bei dem Spiel Schach kann es nur einen Gewinner geben). Wenn es sich jedoch um kein Nullsummenspiel handelt, kann sich das ändern: Würde nämlich beispielsweise Spieler 1 im Fall seiner Zielerreichung die dreifache Belohnung von dem erhalten was Spieler 2 bei seiner individuellen Zielerreichung erhält, so könnten die beiden Spieler kooperieren und vereinbaren, dass Spieler 1 gewinnen soll und die erhaltene Belohnung anschließend geteilt wird, wodurch sich beide Spieler verbessern könnten. Obwohl die Ziele der Spieler also eigentlich von einem Konflikt geprägt sind, ist in diesem Fall Kooperation möglich (im Falle sozial unerwünschter Formen solcher Kooperation würde man von Verschwörung bzw. Absprache sprechen, siehe etwa Kartellbildungen und Preisabsprachen). Abbildung 3.2a veranschaulicht ein rein kompetitives Null-Summen Spiel. In diesem Fall hat Spieler 1 (Zeilenspieler) das Ziel die gleiche Aktion wie Spieler 2 (Spaltenspieler) zu wählen, wohingegen Spieler 2 versucht die gegenteilige Aktion von Spieler 1 zu wählen. Da es sich um ein Nullsummenspiel handelt besteht auch keine Möglichkeit sich durch einen Austausch von Belohnungen beidseitig zu verbessern.

In rein kooperativen Spielen, bei denen die Agenten das gleiche Ziel verfolgen ist kein expliziter Anreiz zu kooperativem Verhalten notwendig. In dieser Kategorie von Spielen können jedoch andere Probleme zwischen den Agenten entstehen, die durch fehlende Koordination bedingt sind. Möchte beispielsweise ein autonom fahrendes Auto ein anderes überholen, so haben beide das Ziel, dass der Überholvorgang ohne Unfall von statten geht, es kann jedoch sein, dass das genaue Manöver für die beiden Akteure unterschiedlich definiert ist und so kann es zu einem Unfall durch Koordinationsprobleme kommen. Diese Situation lässt sich ebenfalls als Matrix-Spiel veranschaulichen und ist in Abbildung 3.2b dargestellt. Das Ziel beider Spieler ist es die gleiche Aktion

zu wählen. Da die Spieler die Aktionswahl des Partners jedoch *a priori* nicht kennen ist ohne eine vorherige Absprache unklar welche Aktion der andere Spieler wählen wird und beide könnten leer ausgehen. Ein Austausch von Belohnungen über einen Markt verspricht hier keine Verbesserung, jedoch sollte der Markt das Problem auch nicht verschärfen.

3.2.6.1 Das Konfliktspiel

Zur Untersuchung der Frage, inwiefern ein Markt in einem kompetitiven bzw. koordinativen Szenario zu einer Verbesserung führt, wird im Folgenden eine empirische Analyse basierend auf einem parametrisierten Matrix-Spiel (dem sogenannten Konfliktspiel) durchgeführt, das zuerst in [76] vorgestellt wurde. Die Idee des Spiels ist es, dass es sich über einen Parameter kontinuierlich von einem rein koordinativen zu einem konfliktären Spiel verwandeln lässt. Somit kann untersucht werden für welchen Grad an Konflikt bzw. Koordination ein Markt eine Verbesserung verspricht. Das Spiel ist in Abbildung 3.3 dargestellt: Beide Spieler haben die Wahl zwischen zwei Aktionen a_1 und a_2, wobei die Wahl von Aktion (a_i, a_j) zu einer Belohnung x, y führt, die in Zelle i, j angegeben ist, wobei x die Belohnung für Spieler 1 (Zeilenspieler) und y die Belohnung für Spieler 2 (Spaltenspieler) ist. Die Höhe der individuellen Belohnungen kann über die Parameter R_1 und R_2 gesetzt werden, so dass sowohl Nullsummenspiele als auch Spiele ohne Nullsummen-Eigenschaft im Fall rein konfliktärer Situationen modelliert werden können.

	a_1	a_2
a_1	$R_1, R_2 - \alpha$	$0, \alpha$
a_2	$0, \alpha$	$R_1, R_2 - \alpha$

Abbildung 3.3: Das Konfliktspiel: Der Konflikt zwischen den Spielern lässt sich über den Parameter α steuern.

Der Grad des Konflikts zwischen den Spielern kann durch den Parameter α gesteuert werden: Für $\alpha = 0$ und mit $R_1, R_2 > 0$ sind die Ziele der Spieler perfekt übereinstimmend, so dass sie sich lediglich einem Koordinationsproblem gegenüber sehen, d.h. sie müssen die gleiche Aktion wählen, um eine positive Belohnung zu erhalten. Für positive Werte von α beginnen die Ziele der Spieler jedoch zu divergieren, da der Spaltenspieler zunehmend einen Anreiz hat, ein unkoordiniertes Ergebnis zu bevorzugen, d.h. wenn Spieler 1 a_1 wählt, bevorzugt Spieler 2 a_2 und umgekehrt. Für $\alpha > R_2$ sind die Ziele der Spieler schließlich völlig widersprüchlich, so dass jeder Spieler nur dann eine positive Belohnung erhält, wenn der andere Spieler das nicht schafft.

3.2.6.2 Evaluation

Zur Evaluation des Konfliktspiels werden die beiden Spieler in Form des Q-Learning Algorithmus modelliert, so wie in Abschnitt 2.2.4 beschrieben. Für das Training wird das Spiel iterativ wiederholt über einen Gesamtzeitraum von 4000 Wiederholungen (Episoden). Jeder Spieler lernt über diesen Zeitraum eine Q-Funktion, aus der die aktuelle Policy abgeleitet wird. Für den Lernprozess werden folgende Hyperparameter verwendet: Die Lernrate wird in allen Durchläufen auf $\alpha = 0,2$ gesetzt und der Diskontierungsfaktor $\gamma = 0,9$. Um exploratives Verhalten der Agenten während des Trainings sicherzustellen, wird das *Epsilon-Greedy*-Verfahren verwendet, wobei ϵ über den Zeitraum der ersten 500 Episoden-Schritte linear abgekühlt wurde beginnend von dem Startwert $\epsilon_{\text{start}} = 1$ bis $\epsilon_{\text{end}} = 0,0001$. Die Belohnungs-Parameter der Agenten werden mit $R_1 = 0,375$ für Agent 1 und $R_2 = 2,625$ für Agent 2 gesetzt, so dass gilt $R_1 + R_2 = 3$.

Um das Konfliktspiel für eine Menge verschiedener Konflikt-Niveaus α zu testen, werden 20 gleichmäßig verteilte Werte für α im Intervall $[0, 3]$ gewählt (für steigendes α nimmt der Konflikt zwischen den Agenten zu). Zu beachten ist, dass sich in Abhängigkeit von α auch die insgesamt erreichbare Gesamtbelohnung im Spiel verändert. So kann für $\alpha = 0$ noch eine Gesamtbelohnung von $R_1 + R_2 = 3$ erzielt werden, für $\alpha = 0,5$ ist jedoch nur noch eine Gesamtbelohnung von $R_1 + R_2 - \alpha = 2,5$ möglich. Die maximal erreichbare Gesamtbelohnung für jedes Niveau von α ist in Abbildung 3.4 mit einer grauen Linie gekennzeichnet und wird als *Pareto-Front* bezeichnet. Die Pareto-Front folgt in diesem Fall einer V-Form, da die mögliche Gesamtbelohnung ab dem Punkt $\alpha = R_1 + R_2/2$ wieder zu steigen beginnt.

Um zu überprüfen, für welche Konflikt-Niveaus ein Markt in diesem Spiel zu einer Verbesserung führt, wurde das Experiment in zwei Varianten durchgeführt: zunächst wurde das Experiment ohne Marktansatz durchgeführt und ein weiteres Mal unter Integration eines bedingten Marktansatzes, wie im vorherigen Abschnitt beschrieben. Alle verwendeten Parameter für das Q-Learning wurden dabei unverändert gelassen. Lediglich der Aktionsraum ändert sich durch die Hinzunahme der Marktaktionen. Beide Varianten wurden insgesamt 20-mal ausgeführt und die Ergebnisse gemittelt. Um nicht den gesamten Trainingsverlauf zu betrachten, wurden lediglich die letzten 500 Wiederholungen des Spiels für die Evaluation herangezogen. Die erzielte Gesamtbelohnung in Abhängigkeit zu dem Konfliktniveau α ist in Abbildung 3.4 zu sehen.

Die Ergebnisse zeigen, dass sowohl im Falle eines integrierten Marktes als auch ohne Markt für den Bereich, in dem das Spiel wenig Konflikt aufweist, bis zu dem Punkt $\alpha = R_1 + R_2/2$ die erzielte Gesamtbelohnung beider Verfahren sich genau auf der Pareto-Front bewegt. Das bedeutet, das sowohl mit als auch ohne Markt in diesem Bereich immer das Optimum erzielt werden kann. Das ist auch plausibel, denn für diesen Bereich ist es für beide Spieler lohnend entweder jeweils Aktion a_1 oder jeweils a_2 zu wählen, d.h. es handelt sich dabei lediglich um ein koordinatives Problem. Den Ergebnissen nach scheint

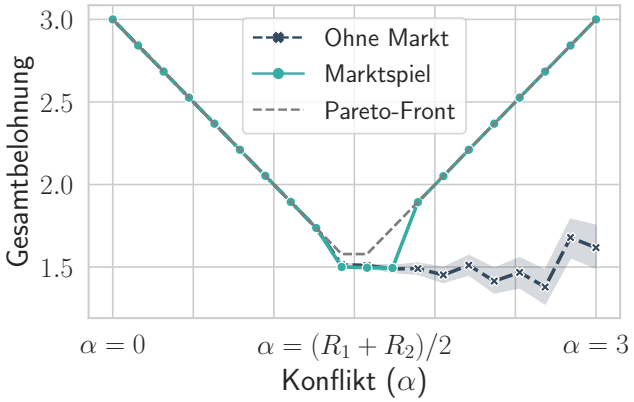


Abbildung 3.4: Vergleich der Ergebnisse im Konfliktspiel mit und ohne Markt.

das koordinative Problem für zwei unabhängig lernende Q-Learning Agenten nicht schwierig zu erlernen zu sein.

Die Verläufe der beiden beginnen sich ab dem Punkt $\alpha = R_1 + R_2/2$ unterschiedlich zu entwickeln, da dies der Punkt ist, an dem das Spiel von einem koordinativen Problem in ein konfliktäres Szenario übergeht. Es zeigt sich, dass der marktbasierte Ansatz nur in dem direkten Übergangsbereich, d.h. um $\alpha = R_1 + R_2/2$, nicht das Gesamtoptimum erreichen kann, im Anschluss aber mit wachsenden α wieder mit der Pareto-Front übereinstimmt. Im Gegensatz dazu liegen die Ergebnisse ohne Markt zunehmend weiter von der Pareto-Front entfernt und auch die Varianz der einzelnen Läufe (kenntlich durch die Konfidenzbänder zum Konfidenzniveau von 95%) nimmt zu.

3.2.7 Zusammenfassung und Diskussion

In diesem Abschnitt wurde ein Konzept zur Erweiterung eines beliebigen stochastischen Spiels um einen Marktmechanismus vorgestellt. Durch die Integration einer Marktfunktion und die Hinzunahme spezifischer Marktaktionen können die Agenten so in einen bidirektionalen Austausch miteinander treten. Zwei spezifische Varianten von Marktfunktionen werden dabei unterschieden: konditionale und bedingungslose Märkte. Zur empirischen Überprüfung in welchen Multiagentensystemen die Einführung eines Marktes sinnvoll sein kann, wurde das Konfliktspiel vorgestellt und eine Evaluation mit zwei lernenden Agenten durchgeführt. Die Ergebnisse aus dem Konfliktspiel zeigen, dass die Gesamtbelohnung durch den Markt insbesondere in dem Bereich eines konfliktären Spiels deutlich verbessert werden kann im Vergleich zu der Situation ohne Markt. Darüber hinaus ist für den Bereich, in dem es sich um ein koordinatives Problem handelt, keine Verschlechterung durch die Hinzunahme des Marktes zu erkennen. Generell ist die Wirkungsweise eines Marktes jedoch abhängig von der konkreten Modellierung. Im Folgenden werden daher weitere

Aspekte diskutiert, die durch das Konzept nicht vollumfänglich definiert sind.

3.2.7.0.1 Generierung von Belohnungen Die Definition der Marktfunktion schreibt nicht vor, in welchem Maße die Agenten Belohnungen miteinander austauschen können. Insbesondere werden keine Einschränkungen bzgl. der Möglichkeit zum Schuldenmachen getroffen. Theoretisch kann ein Agent also einem anderen Agenten eine Belohnung geben, die er selbst gar nicht von der Umgebung erhalten hat, sondern die er selbst generiert hat. Die Tatsache, dass so eine Generierung von Belohnungen zu unterschiedlichen pathologischen Lernverhalten führen kann, wurde bereits in früheren Arbeiten diskutiert. So beschreiben Miller et al. in [54] die Entstehung von Endlosschleifen durch Agenten, die sich gegenseitig durch die Generierung von neuem Geld belohnen. Die Autoren stellen fest, dass diese so genannten Parasiten immer dann auftreten, wenn das System über längere Zeiträume läuft. In den ersten Experimenten zu dieser Arbeit wurden keine Verschuldungsgrenzen gesetzt, wobei festgestellt werden konnte, dass dies das Lernen der Agenten sehr instabil machte, mit einer hohen Varianz der Ergebnisse. Stabilere Lerndynamiken ergeben sich eher dann, wenn es keine Möglichkeit gibt, Belohnungen zu generieren. Das schließt allerdings nicht die Möglichkeit aus, dass Agenten untereinander Schulden machen, in dem bestimmte Käufe erst zu einem späteren Zeitpunkt durch den Transfer einer Belohnung bezahlt werden. In dieser Arbeit werden Schulden allerdings nur episodisch gespeichert, d.h. jede neue Episode startet mit einer kompletten Schulden-Amnestie. Außerdem kann ein Agent seine Schulden gegenüber einem anderen Agenten nur dann begleichen, wenn diese Belohnung auch verdient wurde.

3.2.7.0.2 Preise und Dividenden Die Preise des stochastischen Marktspiels werden exogen festgelegt, d.h. sie werden nicht durch individuelle Verhandlungen unter den Agenten bestimmt oder basierend auf Angebot und Nachfrage ermittelt. D.h. die Agenten machen lediglich Kauf- und Verkaufsangebote und wenn eine Transaktion zustande kommt, wird diese durch einen festen Wert p entlohnt. Dieser Ansatz ermöglicht es allen Agenten unter den gleichen Bedingungen zu kaufen und zu verkaufen, und unterscheidet sich daher von der für die Hayek-Maschine [10] vorgeschlagenen Gebotszuweisungsstrategie, bei der den Agenten ein festes, zufälliges Gebot zugewiesen wird, so dass nur Agenten mit geeigneten Geboten überleben. Die Wahl des Preises ist somit Teil der zu definierenden Hyperparameter und spielt eine Rolle dafür, ob der Markt zu einem sinnvollen Austausch genutzt werden kann. Die Frage wie ein Verhandlungsprozess zwischen zwei Agenten zum Finden eines Preises modelliert werden kann wird in dem zweiten Abschnitt (Abschnitt 3.3) dieses Kapitels betrachtet.

3.2.7.0.3 Der Markt als Teil der Umgebung Die Integration eines Marktes erfordert, dass die Evaluations-Umgebung um eine Marktfunktion erweitert

werden kann. Der Markt ist somit Teil der Umgebung und kann daher nicht von einzelnen Teilnehmern manipuliert werden. Betrug ist somit ausgeschlossen. Dieses Vorgehen orientiert sich an dem Konzept idealisierter Märkte [55], die Regeln definieren, die es den einzelnen Teilnehmern unmöglich machen, Eigentumsrechte zu verletzen, andere Agenten zu täuschen oder zu betrügen, da diese Gesetze von der Umwelt durchgesetzt werden.

3.3 Verhandlungsprozesse

Im vorangegangenen Abschnitt wurde das Konzept des stochastischen Marktspiels vorgestellt, das es den Agenten ermöglicht, Belohnungen durch bidirektionale Transaktionen auszutauschen. Dabei wurde die Höhe des Austausches, also der Transaktionswert, als exogener Parameter festgelegt. Es schließt sich die Frage an, wie sich ein Verhandlungsprozess zur Findung eines Transaktionswertes mittels autonomer KI-Agenten modellieren lässt. Verhandlungsprozesse werden auch in der Spieltheorie behandelt. Eine Verhandlungssituation ist dort als ein Szenario beschrieben bei dem die Individuen die Möglichkeit haben zu kollaborieren, wodurch alle profitieren würden, allerdings gibt es unterschiedliche Möglichkeiten wie sich die Zusammenarbeit gestalten kann, so dass es Uneinigkeit darüber gibt, welche konkrete Form der Kollaboration gewählt werden soll [59].

In diesem Abschnitt werden nun Möglichkeiten zur Modellierung von Verhandlungsprozessen zwischen zwei Spielern mittels *deep* Multi-Agent Reinforcement Learning vorgestellt. So ein Verhandlungsprozess kann beispielsweise das Finden eines gleichgewichtigen Preises darstellen. Da die Preisfindung ein zentraler Bestandteil für die Funktionsweise der in Abschnitt 3.2 vorgestellten Marktansätze darstellt, könnte ein solcher Prozess potentiell dazu eingesetzt werden. Treten zwei autonome KI-Systeme in einen Verhandlungsprozess miteinander, so stellt sich zuerst die Frage, in welcher Form der Verhandlungsprozess repräsentiert wird, also wie die Agenten die Zustände dieses Prozesses wahrnehmen. Dazu werden in diesem Abschnitt zwei Möglichkeiten zur Darstellung einer Verhandlungssituation vorgestellt, die das Training zweier Reinforcement Learning Agenten ermöglichen. Die erste Domäne modelliert die Situation zweier Verhandlungspartner, die versuchen, ein gemeinsames Gut aufzuteilen, wie beispielsweise eines bestimmten Geldbetrages. Die zweite Domäne behandelt ein Verkäufer-Käufer-Szenario, d.h. ein Agent tritt in der Rolle des Verkäufers und ein Agent tritt als Käufer auf. Ziel ist es, dass die Agenten sich auf einen Preis für ein Verkaufsgut einigen.

Neben der Frage der Modellierung des Prozesses, so dass er von selbstlernenden Systemen verarbeitet werden kann, gilt es auch zu analysieren, welche spezifischen Aspekte das erzielte Verhandlungsergebnis beeinflussen. Hierbei können sowohl Agenten-interne Aspekte relevant sein als auch Parameter, die nicht dem Agenten zugeordnet werden, sondern als exogene Attribute gelten. Für den Einsatz in echten Anwendungen ist es beispielsweise wichtig zu wissen,

welche Aspekte das Verhandlungsergebnis für eine der Parteien positiv bzw. negativ beeinflussen können. Ein dafür relevanter Aspekt aus spieltheoretischer Sicht ist die Risikoaversion eines Agenten [72]. Zur Modellierung risikoaverser Agenten, wird daher eine spezielle Agentenarchitektur vorgestellt, die es erlaubt den Grad der Risikoaversion des Agenten über einen einzelnen Parameter zu steuern. Das Ziel der dazugehörigen Experimente ist der empirische Nachweis, dass die axiomatischen Modelle der Spieltheorie zum Einfluss von Risiko in Verhandlungssituationen auch durch das Training mittels Reinforcement Learning validiert werden können. Der zweite Aspekt zu den Einflüssen in Bezug auf das erzielte Verhandlungsergebnis ist der Grad der Informationen, der den Agenten jeweils zur Verfügung steht. Es ist bekannt, dass ein zu hoher Grad an Informationsasymmetrie zwischen Verhandlungspartnern zu Marktversagen führen kann, da Käufer die angebotenen Güter nicht mehr mit gleicher Sicherheit bewerten können wie Verkäufer [2]. Der Einfluss der Informationsasymmetrie wird daher in einem Experiment untersucht, bei dem die Agenten unterschiedliche Informationen über ein zu verhandelndes Gut erhalten. Es zeigt sich, dass ein höherer Grad an Informationsasymmetrie sich negativ auf den Verhandlungserfolg auswirkt. Auch hier lassen sich also die spieltheoretischen Vorhersagen mittels Reinforcement Learning empirisch validieren.

3.3.1 Verwandte Arbeiten

Die Untersuchung von Verhandlungsprozessen zwischen eigennützig handelnden Akteuren ist klassischerweise Gegenstand der Spieltheorie [59, 72, 73]. Unter der Annahme, dass die Spieler vollkommen rational sind, wird versucht Gleichgewichte und dominante Strategien zu ermitteln. Darüber hinaus wird davon ausgegangen, dass die Spieler eine dominante Strategie spielen, wenn diese bekannt ist. Dabei wird von verschiedenen Aspekten realer Verhandlungssituationen abstrahiert, wie beispielsweise spezifischen Fähigkeiten (Verhandlungsgeschick) oder Eigenschaften (Risikoaversion) der Spieler. Bei lernenden Agenten kann weder davon ausgegangen werden, dass dominante Strategien *a priori* bekannt sind, noch dass, diese selbst nachdem sie erlernt wurden, immer auch gespielt werden (siehe Exploration vs. Exploitation), da die Agenten Aktionen gemäß einer Wahrscheinlichkeitsverteilung wählen und somit auch immer wieder suboptimale Aktionen ausführen. Diese Agenten werden daher als begrenzt rational (engl. *boundedly rational* bezeichnet.

Die evolutionäre Spieltheorie lässt die Annahme perfekt rationaler Agenten fallen. Strategien entwickeln sich nach dem evolutionären Prinzip des *Survival of the Fittest*, so dass sich die (relativ) überlebensfähigsten Strategien durchsetzen. Dabei wird die relative Fitness von Strategien, durch die wiederholte Interaktion von Populationen unterschiedlicher Strategien, miteinander ermittelt, so dass sich im Verlauf fittere Strategien durchsetzen. Ellingsen untersucht die Entstehung von Verhandlungsverhalten in bilateralen Verhandlungssitua-

tionen [23]. Dabei gibt es zwei Klassen von Agententypen: Die Agenten sind entweder hartnäckig, d.h. ihre Forderungen sind unabhängig vom Gegner, oder sie sind adaptive Agenten, die sich an das erwartete Spiel ihres Gegners anpassen. Es zeigt sich, dass hartnäckige Agenten einen evolutionären Vorteil haben, solange die Verhandlungsmasse sicher ist, d.h. beiden Seiten ohne Unsicherheit bekannt ist. Eine Arbeit die sich mit dem Thema Unsicherheit in einem Verkäufer- (informiert) und Käufer- (uninformiert) Szenario befasst, ist von Konrad und Morath veröffentlicht worden [41]. Die Autoren zeigen, dass sich unvollständige Informationen negativ auf den Handel zweier Parteien auswirken können, wenn die Spieler evolutionär stabile Strategien anwenden, verglichen mit dem entsprechenden perfekten Bayes-Gleichgewicht.

Die Evolution von Fairness wird in [69] behandelt. Die Autoren zeigen, dass durch natürliche Selektion Fairness im anonymen One-Shot-Ultimatum-Spiel begünstigt wird, wenn Agenten Fehler bei der Beurteilung der Auszahlungen und Strategien anderer machen. In [26] wird die kompetitive Koevolution in einer Umgebung mit unvollständigen Informationen untersucht. Die Autoren verwenden ein Verkäufer-Käufer-Szenario und vergleichen ihre Ergebnisse mit denen, die aus spieltheoretischen Analysen hervorgehen. Es wird gezeigt, dass der von genetischen Algorithmen gefundene stabile Zustand nicht immer mit dem spieltheoretischen Gleichgewicht übereinstimmt. Darüber hinaus hängt das stabile Ergebnis von der Ausgangspopulation ab, da sich die Spieler gegenseitig an die Strategie des anderen anpassen.

Die Modellierungen der evolutionären Spieltheorie eines Verhandlungsprozesses kommen dem in dieser Arbeit verfolgten Ansatz näher, da sich hier Agenten aneinander durch bestärkendes Lernen anpassen und sich Strategien somit emergent über den Verlauf vieler Trainingsepisoden bilden. Im Gegensatz zu Ansätzen aus der evolutionären Spieltheorie werden in diesem Abschnitt jedoch nicht die evolutionären Dynamiken der Strategien betrachtet. Der Fokus liegt mehr darauf, die sich aus agentenspezifischen Parametern, wie Risikoaversion oder Informationsasymmetrie ergebenden Ergebnisse, zu analysieren.

3.3.2 Konzeption zweier Verhandlungs-Domänen

In diesem Abschnitt werden zwei Lernumgebungen vorgestellt, die zum Zwecke der Modellierung bilateraler Verhandlungsprozesse für unabhängig lernende Agenten entwickelt wurden. Um einen iterativen Verhandlungsprozess durch zwei *deep* Reinforcement Learning Agenten lernbar zu machen, wurden die Domänen so entwickelt, dass die Agenten die Verhandlung durch Modifikationen ihrer Umgebung steuern können. Der Verhandlungsprozess gestaltet sich somit durch die Veränderung der Umgebung durch die Aktionen der Agenten.

Die Verhandlungsszenarien werden durch zweidimensionale Gitter-Welten repräsentiert, deren Breite und Höhe variiert werden kann. Um den Verhandlungsprozess zu steuern, bewegen sich die Agenten durch das Gitter, indem sie ihre Aktionen $a \in \mathcal{A}$ anwenden. Durch das Überschreiten von einzelnen Zellen

modifizieren die Agenten die Umgebung, wodurch sich das aktuelle Verhandlungsergebnis verändert. Jeder Zustand $s \in \mathcal{S}$ in dem sich die Umgebung befinden kann, stellt also ein bestimmtes Verhandlungsergebnis dar, das die Agenten durch die Wahl ihrer Aktionen $a \in \mathcal{A}$ beeinflussen können. Der Verhandlungsprozess wird also durch die Folge von Aktionen während einer Episode spezifiziert. Der Zustand s_{final}, mit dem eine Episode endet, definiert dann das endgültige Verhandlungsergebnis, woraus sich die resultierenden Belohnungen der Agenten ableiten. Die Belohnungen spiegeln dabei den Verhandlungserfolg wider, geben also an, wie gut ein Agent eine Verhandlung führen konnte. Mit jeder gespielten Episode wird eine Verhandlung komplett abgeschlossen. Somit lernen die Agenten das Verhandeln über den gesamten Trainingsverlauf, d.h. über den Zeitraum vieler gespielter Episoden. Die Domänen unterscheiden sich darüber hinaus bezüglich der Symmetrie der Aufgaben für die Agenten. Die erste Domäne ist hinsichtlich der Aufgabe für beide Agenten identisch, d.h. beide Agenten sehen sich vor der gleichen Aufgabe. Die zweite Domäne hat eine asymmetrische Aufgabenstellung, so dass die Agenten innerhalb des Verhandlungsprozesses unterschiedliche Rolle einnehmen und unterschiedliche Belohnungen erhalten.

Die Verhandlungsdomänen bieten die Grundlage um zu analysieren, welche Faktoren lernender Algorithmen das Verhandlungsergebnis zwischen zwei Agenten beeinflussen können. Dabei können die Agenten auch durch unterschiedliche Klassen von Algorithmen repräsentiert werden, da die Schnittstelle lediglich die Wahl einer Aktion a_t für jeden Zeitschritt t des Verhandlungsprozesses erfordert. Doch auch unter Verwendung gleicher Algorithmen kann es Einflussfaktoren geben, wie beispielsweise die Belegung einzelner Hyperparameter (Lernrate oder Explorationsstrategie), die das Ergebnis beeinflussen können. Im Folgenden werden die beiden konzeptionierten Verhandlungsdomänen vorgestellt.

3.3.2.1 Domäne zur Modellierung eines Teilungsprozesses

Eine Form der Verhandlung, die oft Gegenstand spieltheoretischer Analysen ist, ist die Aufteilung eines Wertgegenstandes zwischen mehreren Verhandlungsparteien [5, 13]. In diesem Szenario versuchen die Verhandlungspartner einen Geldwert untereinander aufzuteilen, wobei jeder Akteur einen höheren Anteil gegenüber einem niedrigeren Anteil bevorzugt. Der Geldwert kann dabei in diskreten Einheiten (z.B. Cents) aufgeteilt werden. Ein solcher Verhandlungsprozess, in Form einer Multiagenten-Domäne, wird im Folgenden vorgestellt. Die Domäne hat den Namen *Divide-The-Grid* in Anlehnung an das spieltheoretische Modell mit dem Namen *Divide-The-Dollar*. Eine Visualisierung der Domäne ist in Abbildung 3.5 dargestellt. Es gibt zwei Agenten, dargestellt durch den roten und den blauen Kasten, die sich mittels ihrer vier möglichen Aktionen $(\leftarrow, \rightarrow, \uparrow, \downarrow)$ durch das Feld bewegen können. Zu Beginn jeder Episode starten beide Agenten in zwei gegenüberliegenden Ecken des Feldes. Indem die Agenten sich durch das Feld bewegen und über eine Zelle (i, j)

gehen, beanspruchen sie diese Zelle als Teil ihres Verhandlungsergebnisses für sich (in Abbildung 3.5 dargestellt durch den farbig transparenten Bereich).

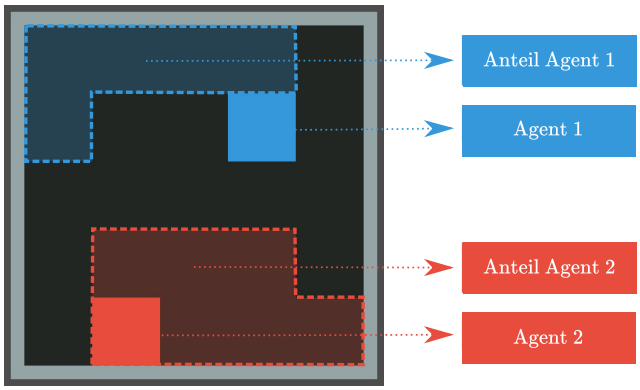


Abbildung 3.5: Die Verhandlungsdomäne *Divide-The-Grid* modelliert die Aufteilung der Zellen einer Grid-Welt zwischen zwei Agenten.

Am Ende einer Episode wird das Verhandlungsergebnis durch die Anzahl der Zellen bestimmt, die jeder Agent im Verlauf einer Episode für sich beanspruchen konnte. Die Belohnung der Agenten berechnet sich aus der Anzahl gesammelter Zellen: ist C die Gesamtzahl an Zellen im Spiel, so erhält Spieler i eine Belohnung die proportional zum Anteil seiner gesammelten Zellen $\frac{n_i}{C}$ ist, wobei n_i die Anzahl gesammelter Zellen von i ist. Die spieltheoretische Definition eines Verhandlungsprozesses gibt stets die Möglichkeit für ein Veto vor, d.h. jeder Agent hat die Möglichkeit den Verhandlungsprozess zu beenden, wodurch das status-quo Ergebnis realisiert wird. Dieser Aspekt wurde in *Divide-The-Grid* dadurch berücksichtigt, dass Agenten die Möglichkeit haben, den Prozess abzubrechen, in dem sie eine bereits von dem anderen Agenten beanspruchte Zelle für sich beanspruchen. In diesem Fall endet die Episode und beide Agenten erhalten die status-quo Belohnung, die das Scheitern der Verhandlung kennzeichnet. Falls kein Agent die Verhandlung frühzeitig beendet, endet eine Episode nach einer festen Anzahl an Zeitschritten.

3.3.2.2 Domäne zur Modellierung einer Käufer-Verkäufer Verhandlung

Neben Aufteilungs-Verhandlungen wie im vorangegangenen Abschnitt beschrieben, ist ein weiteres relevantes Verhandlungsszenario eine Verkaufs-Verhandlung zwischen einem Verkäufer eines Gutes an einen Käufer. Da sich sowohl auf Verkaufs- als auch auf der Kaufseite jeweils nur ein Akteur befindet, wird dieses Szenario auch als bilaterales Monopol bezeichnet. Das Verhandlungsproblem der beiden Parteien besteht darin, einen Preis für einen Verkaufsgegenstand (Handelsgut) festzulegen, wobei der Verkäufer einen möglichst hohen Preis erzielen möchte und der Käufer einen möglichst niedrigen Preis präferiert. Beide Parteien versuchen einen Preis zu finden, der gerade noch im Akzeptanz-Bereich der anderen Partei liegt, so dass der eigene Netto-Nutzen durch den Verkauf bzw. den Erwerb des Handelsgutes maximal ist.

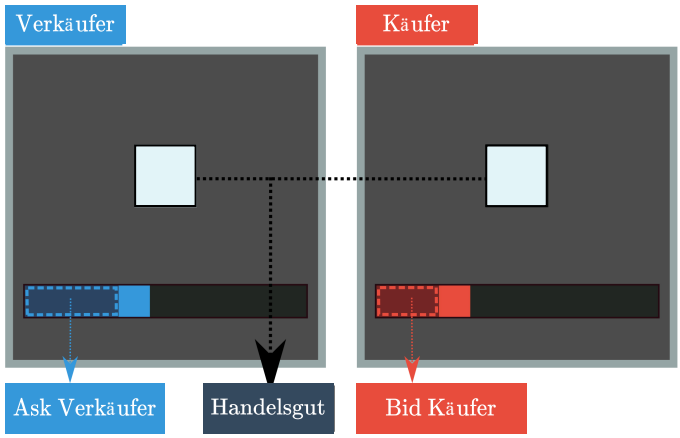


Abbildung 3.6: Käufer-Verkäufer-Verhandlungs-Domäne: Zwei Agenten müssen einen Preis festlegen, um ein Handelsgut zu transferieren.

Auch diese Situation lässt sich durch eine Grid-Welt repräsentieren. Da der Käufer-Agent und der Verkäufer-Agent jeweils ihren Preis definieren müssen, hat jeder Agent sein eigenes Feld, sowie in Abbildung 3.6 dargestellt. Die Agenten sind wieder als farbige Vierecke gekennzeichnet (blau = Verkäufer, rot = Käufer) und das Handelsgut als größeres Viereck in der Mitte des Felds. Unterschiedliche Handelsgüter können über verschiedene Farben gekennzeichnet werden. Die Agenten können ihren Verkaufs- bzw. Kaufpreis nun in folgender Art und Weise angeben: die Agenten können sich horizontal nach rechts oder links bewegen. Gehen sie nach rechts, so erhöhen sie ihr Angebot, gehen sie nach links, so reduzieren sie ihr Angebot, d.h. je weiter sich der Agent im Verlauf einer Episode nach rechts bewegt, desto höher ist am Schluss der geforderte Preis im Falle des Verkäufers bzw. der akzeptierte Preis im Falle des Käufers. Die aktuellen geforderten Preise sind durch den farbig-schattierten Bereich im Balken des Agenten gekennzeichnet (siehe Detailansicht in Abbildung 3.7).

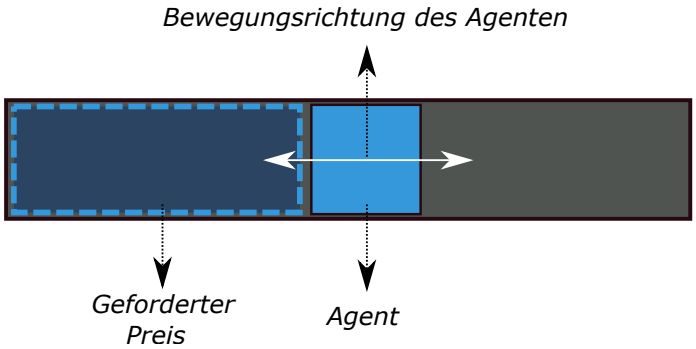


Abbildung 3.7: Detailansicht des Verkäufer-Agenten in der Verkäufer-Käufer-Domäne.

Damit ein erfolgreicher Transfer am Ende der Episode stattfindet, das Handelsgut also tatsächlich verkauft wird, muss der Angebotspreis (Bid-Preis) min-

destens so hoch sein wie der Nachfragepreis (Ask-Preis), d.h. es muss gelten $p_b \geq p_s$. Es besteht keine Möglichkeit für die Agenten die Verhandlung frühzeitig zu beenden, d.h. Episoden haben eine feste Länge von T Schritten, nach denen die Verhandlung endet. Sei p der tatsächlich realisierte Preis, dann entspricht die Belohnung des Verkäufers dem erzielten Preis abzüglich seines initialen Nutzens aus dem Besitz des Gutes, also $R_s = p - U_s$ und die Belohnung des Käufers entspricht dem Nutzen aus dem Erwerb des Gutes U_b abzüglich des gezahlten Preises, also $R_b = U_b - p$ (es wird davon ausgegangen, dass der Käufer keinen Nutzen aus dem Nicht-Erwerb des Gutes bezieht).

3.3.2.2.1 Informationsasymmetrie In einer Käufer-Verkäufer Situation kann davon ausgegangen werden, dass der Verkäufer und der Käufer unterschiedliche Informationen in Bezug auf das gehandelte Gut haben. In diesem Fall spricht man von Informationsasymmetrie zwischen den Parteien. Informationsasymmetrie wird als eine der zentralen Hemmnisse für effizienten Handel beschrieben [3]. Akerlof [3] formulierte die Theorie des *Market for Lemons*, wonach asymmetrische Informationen sogar zu einem kompletten Zusammenbruch wirtschaftlichen Austauschs führen können. Akerlof beschreibt diesen Effekt anhand eines Gebrauchtwagen-Marktes: nimmt man an, es gibt einen Markt für Gebrauchtwagen in dem es gute Gebrauchtwagen und schlechte Gebrauchtwagen (sogenannte *Lemons*) gibt und Käufer darüber hinaus die Qualität eines Gebrauchtwagens nur schwer überprüfen können. Dann werden die Käufer lediglich bereit sein einen Preis zu zahlen, der das Risiko berücksichtigt unbeabsichtigt einen schlechten Gebrauchtwagen zu kaufen. Dieser Preis wird daher niedriger sein als der Preis, den der Verkäufer eines guten Gebrauchtwagens bereit ist zu akzeptieren. Als Konsequenz werden sich also Verkäufer guter Gebrauchtwagen aus dem Markt zurückziehen, so dass nur noch Verkäufer schlechter Gebrauchtwagen (sogenannter *Lemons*) im Markt verbleiben. Da niemand einen schlechten Gebrauchtwagen kaufen will, bricht der Markt schlussendlich komplett zusammen. [3]

Zur Untersuchung des Einflusses von Informationsasymmetrie auf den Verhandlungserfolg in einer Verkaufssituation, lässt sich die Verkäufer-Käufer Domäne folgendermaßen verwenden: Es gibt zwei Handelsgüter, die gehandelt werden und die in zwei unterschiedlichen Qualitätsklassen vorkommen können. Die Beobachtung des Verkäufers umfasst dann auch die Information über die Qualität des aktuellen Handelsgutes, wohingegen der Käufer die Qualität nicht beobachten kann.

3.3.3 Modellierung risikoaverser Agenten

In der axiomatischen Verhandlungstheorie wird Risikoaversion durch konkave Nutzenfunktionen modelliert [72]. Das bedeutet, dass ein Spieler eine sichere Auszahlung in Höhe von $0,5$ Geldeinheiten einer Lotterie vorzieht, in der er mit einer Wahrscheinlichkeit von $0,5$ eine Auszahlung in Höhe von 1 Geldein-

heit bekommt und ebenfalls mit Wahrscheinlichkeit $0,5$ eine Auszahlung von 0 Geldeinheiten. Für ein risikoneutrales Individuum spielt es keine Rolle, ob es die Lotterie oder die sichere Auszahlung erhält, da beide den gleichen Erwartungswert haben. Eine zentrale Erkenntnis aus der axiomatischen Verhandlungstheorie ist jedoch, dass der risikoaversere Spieler auch einen geringeren Anteil von der Verhandlungsmenge erhält [72].

Um den Einfluss der Risikobereitschaft auf das Ergebnis eines Verhandlungsprozesses zu untersuchen, wird im Folgenden eine Agentenarchitektur vorgestellt, die es erlaubt die Risikoaversion eines Agenten mittels eines Parameters zu steuern. Die Architektur beruht auf dem von Bellemare et al. vorgeschlagen Ansatz zum Erlernen der Werteverteilung (engl. *Value-Distribution*) anstelle der Wertefunktion [12]. Dabei wird für jedes Zustand-Aktions-Paar (s, a) die gesamte Werte-Verteilung approximiert, anstelle eines skalaren Funktionswertes, wie er durch die Q-Funktion normalerweise abgebildet wird. Wenn die gesamte Wertverteilung bekannt ist, können andere Metriken für die Entscheidungsfindung angewendet werden, die aus dieser Verteilung abgeleitet werden können, wie beispielsweise das Treffen von sicheren oder risikoaversen Entscheidungen. Anstatt eine Entscheidung anhand der Q-Werte abzuleiten, stehen dem Agenten bei verteilungsbasiertem Lernen für jeden Zustand $s \in \mathcal{S}$ jeweils eine Wahrscheinlichkeitsverteilung über den zu erwartenden Erlös für jede Aktion $a \in \mathcal{A}$ zur Verfügung. Eine Visualisierung der Wahrscheinlichkeitsverteilungen, für die vier Aktionen eines gegebenen Zustands der *Divide-The-Grid*-Domäne, ist in Abbildung 3.8 zu sehen (hier aus Sicht des blauen Agenten).

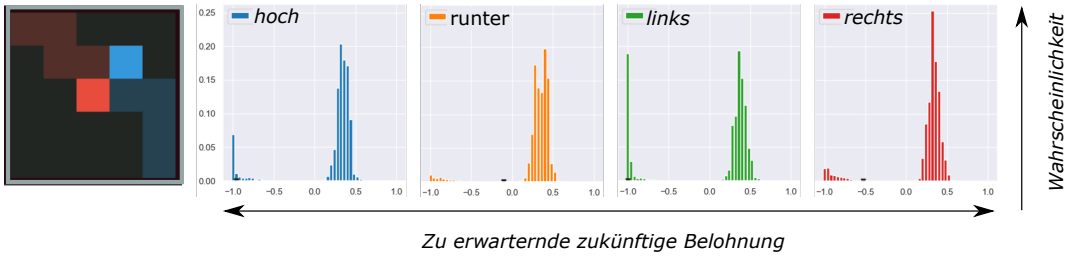


Abbildung 3.8: Verteilungen über die zu erwartenden zukünftigen Belohnungen, für die vier möglichen Aktionen des blauen Agenten, in dem dargestellten Zustand.

Da der Agent in dieser Umgebung über vier mögliche Aktionen (*hoch, runter, links, rechts*) verfügt ergeben sich vier mögliche Wahrscheinlichkeitsverteilungen. In diesem Beispiel ist zu erkennen, dass die Ausführung der Aktion *links*, laut der gelernten Verteilung, einen größeren Teil der Wahrscheinlichkeitsmasse für negative Werte aufweist. Das liegt daran, dass in dem links in der Abbildung dargestellten Zustand, ein Schritt nach links des blauen Agenten, zu einem Abbruch der Verhandlung führen würde, da in diesem Fall der blaue Agent, eine bereits von dem roten Agenten geforderte Zelle, beansprucht. Die Spitze im negativen Bereich der Verteilung der Aktion *links*, kennzeichnet

somit das Ergebnis für eine nicht erfolgreiche Verhandlung, deren Ergebnis in diesem Beispiel -1 beträgt.

Standardmäßig treffen Agenten, die mittels Reinforcement Learning trainiert werden, risikoneutrale Entscheidungen, da als Entscheidungskriterium die zu erwartende zukünftige Belohnung verwendet wird. Dadurch werden mögliche extreme Ausreißer des Belohnungssignals, durch dieses Kriterium in der Entscheidung des Agenten, nicht berücksichtigt. So können sich beispielsweise hohe positive Belohnungen und hohe negative Belohnungen im Mittel aufheben, so dass die Streuung der Werte für die Entscheidung unerheblich bleibt. Soll der Agent risikoaverses Verhalten zeigen, so ist der Erwartungswert als Kriterium unzureichend, da die Möglichkeiten über potentiell hohe Verluste in einzelnen Fällen darin nicht zwangsläufig abgebildet werden.

Eine Möglichkeit risikoaverse Entscheidungen zu treffen, beruht auf der Idee, die gesamte Verteilung der zu erwartenden zukünftigen Belohnungen des Agenten als Grundlage für die Entscheidung zu verwenden. Ist diese Verteilung bekannt, so kann die Wahl der Aktion in einer Art und Weise erfolgen, die es ermöglicht, potentiell eintretende hohe Verluste (negativen Belohnungen) zu berücksichtigen. So kann der blaue Agent in Abbildung 3.8 unter Berücksichtigung der Werteverteilungen aller Aktionen feststellen, dass die Aktion *links* eine höhere Wahrscheinlichkeit für eine negative Belohnung von -1 aufweist, wohingegen die Aktion *rechts* nur eine geringe Wahrscheinlichkeit für negative Belohnungen zeigt. Auch wenn die Aktionen einen ähnlichen Erwartungswert haben, würde ein risikoaverser Entscheidungsträger in diesem Fall die Aktion *rechts* der Aktion *links* vorziehen.

3.3.3.1 Conditional Value at Risk

Um nun eine risikoaverse Entscheidung gemäß der Werteverteilungen, über alle Aktionen eines RL-Agenten zu ermöglichen, lassen sich unterschiedliche Kriterien verwenden [28]. In dieser Arbeit wird dafür die Risikometrik *Conditional Value at Risk* ($CVaR$) eingesetzt, ein Kriterium, das insbesondere aus der Portfolio-Optimierung bekannt ist [71]. Die Definition von $CVaR$ baut auf dem Value at Risk (VaR) auf.

Value at Risk VaR_α ist definiert über eine Zufallsvariable X, die Verlust darstellt (d.h. negative Werte von X entsprechen positiven Belohnungen), sowie einem Konfidenzparameter $0 < \alpha < 1$ und wird folgendermaßen berechnet: [38]:

$$VaR_\alpha(X) := \min\{c : P(X \leq c) \geq \alpha\}$$

Einfach ausgedrückt, lässt sich VaR als der minimale Verlust in den $1 - \alpha \times 100\%$ schlimmsten Fällen interpretieren. Ein bekannter Nachteil von VaR ist, dass darin keine weiteren Informationen über die Verteilung der Ereignisse in den $1-\alpha \times 100\%$ schlimmsten Fällen enthalten sind, was bei hohen potenziellen Verlusten wichtig sein könnte. Um auch die Verteilung dieser potentiellen Verluste in die Entscheidungsfindung einfließen zu lassen, wurde VaR entsprechend

erweitert. Die Erweiterung dazu ist als Conditional Value at Risk bekannt und ist folgendermaßen definiert [38]:

$$CVaR_\alpha(X) := \mathbb{E}[X|X \geq VaR_\alpha(X)]$$

wobei X eine kontinuierliche Zufallsvariable ist, die Verlust repräsentiert und $0 < \alpha < 1$ ist der Konfidenz-Parameter, der den Anteil der zu berücksichtigenden schlimmsten Verluste definiert. CVaR lässt sich also als erwarteter Wert in den $1 - \alpha \times 100\%$ schlimmsten Fällen interpretieren.

3.3.3.2 Risikoaverse Agenten

Auf Basis des verteilungsbasierten Reinforcement Learning nach Bellemare et al. [12], lassen sich risikoaverse Agenten, unter Verwendung von Conditional Value at Risk als Risikometrik, implementieren. Der Grad der Risikoaversion kann dabei durch den Parameter α gesteuert werden. Ein risikoaverser Agent bevorzugt dann Aktionen mit höherem *CVaR*-Wert zu einem gegebenen α (wenn die Variable nicht Verlust, sondern Belohnung darstellt). Durch Verringerung des Konfidenzparameters α wird der Anteil berücksichtigter hoher Verluste (niedriger Belohnungen) kleiner, so dass die Aktionswahl zunehmend empfindlicher auf einzelne hohe Verluste (niedrige Belohnungen) reagiert, der Agent wird dadurch also risikoaverser. Um zu gewährleisten, dass der Agent auch unter risikoaverser Aktionswahl kontinuierlich exploriert, kann dieser Ansatz ebenfalls mit einer ϵ-*greedy* Explorationsstrategie kombiniert werden, wobei der Agent mit Wahrscheinlichkeit ϵ eine zufällige Aktion wählt und mit Gegenwahrscheinlichkeit $1 - \epsilon$ die Aktion mit dem höchsten *CVaR*-Wert.

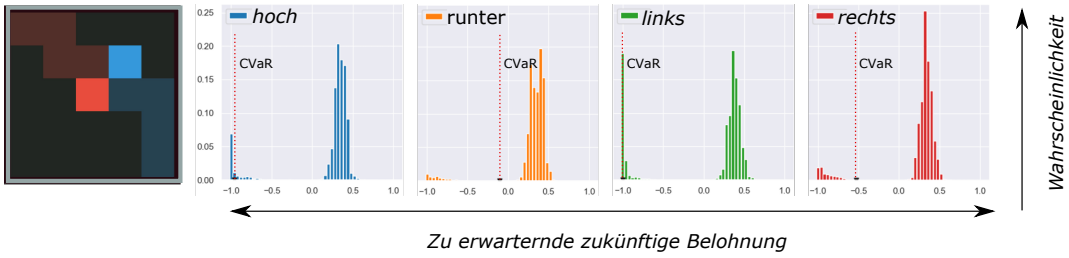


Abbildung 3.9: Verteilungen über die zu erwartenden zukünftigen Belohnungen, für die vier möglichen Aktionen des blauen Agenten, inklusive *CVaR*-Werte für alle Verteilungen.

Zur Veranschaulichung der *CVaR*-Werte für die Situation in der Verhandlungsdomäne *Divide-The-Grid*, sind die Verteilungen und *CVaR*-Werte in Abbildung 3.9 dargestellt. Die *CVaR*-Werte zu jeder Aktion lassen die Sensitivität dieser Metrik zu den potentiell negativen erwarteten Belohnungen erkennen. So spiegelt sich das hohe Risiko eines vorzeitigen Verhandlungsabbruchs durch die Wahl der Aktion *links* in einem geringen *CVaR*-Wert wider, wohingegen Aktionen mit geringem Risiko einen höheren *CVaR*-Wert aufweisen.

3.3.4 Experimente

In diesem Abschnitt werden nun zwei Experimente zu den beiden Verhandlungsdomänen vorgestellt. Das erste Experiment untersucht dabei die Auswirkungen der Risikoaversion eines Agenten auf dessen Anteil des Verhandlungsergebnisses. In dem zweiten Experiment wird der Einfluss von asymmetrischen Informationen zwischen den Agenten betrachtet, wobei die Frage ist, inwiefern sich eine Informationsschieflage negativ auf den generellen Verhandlungserfolg auswirkt.

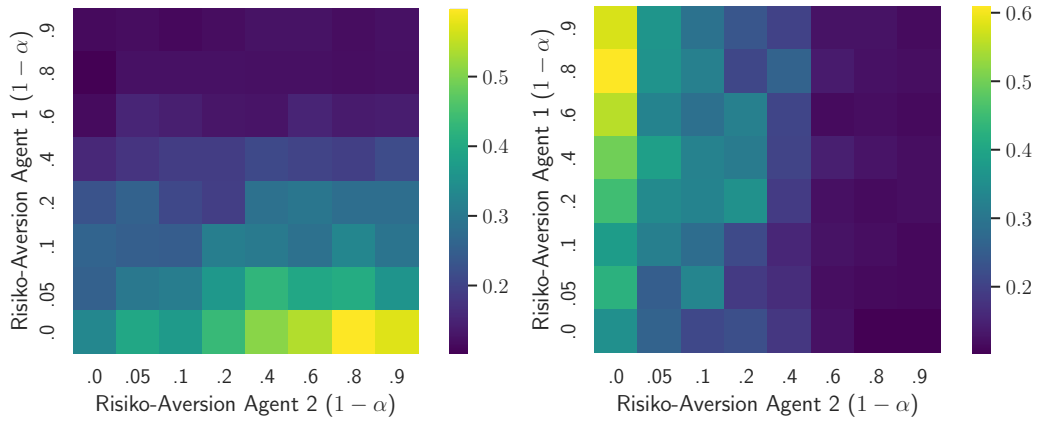
Für beide Experimente werden Agenten verwendet, die auf der in Abschnitt 2.2.7 vorgestellten *Categorical DQN*-Architektur [12] beruhen. Jeder Agent wird dabei in Form eines *convolutional neural network* repräsentiert. Die Agenten lernen auf Basis von Bilddaten der Größe $(84, 84, 3)$ (Breite, Höhe, Anzahl Farbkanäle). Die Netzarchitektur besteht aus drei convoluational Layern, wobei die Layer kleiner werdende Filterkernel (Breite, Höhe) der Größe $(8, 8)$ (Layer 1), $(4, 4)$ (Layer 2) und $(3, 3)$ (Layer 3) besitzen. Die Filter werden mit einer Schrittgröße von 4 (Layer 1), 2 (Layer 2) bzw. 1 (Layer 3) über die Eingangsdaten bewegt. Die Ausgangsdimension des ersten Layers beträgt 32 und 64 für Layer 2 und Layer 3. Nach jedem convolutional Layer wird die *Rectified Linear Unit*-Aktivierung angewendet. Nach den convolutional Layern wird der 64-dimensionale Tensor zu einem eindimensionalen Array der Länge 512 transformiert und an einen *Dense Layer* übergeben, d.h. eine Schicht, bei der jedes Neuron mit jedem Neuron der vorausgehenden Schicht verbunden ist. Schlussendlich wird in der letzten Schicht die Anzahl der Ausgänge durch eine weitere Anwendung eines *Dense Layer* in ein Array der Länge 4 (entspricht Anzahl der möglichen Aktionen des Agenten) transformiert und final durch eine lineare Aktivierungsfunktion geführt.

Um kontinuierliche Exploration der Agenten zu gewährleisten, wird für die Verhaltenspolicy des Agenten, d.h. die Policy mit der der Agent seine Trainingsdaten generiert, das Epsilon-Greedy Verfahren verwendet. Die Explorationsrate wird dabei linear über die Trainingszeit abgekühlt, beginnend mit einer Explorationsrate von $\epsilon = 0,3$ bis zu einer minimalen Rate von $\epsilon = 0,001$. Während des Trainings wird die *Target-Policy* alle 100 Zeitschritte auf den Stand des aktuellen Q-Netzwerks des Agenten gesetzt.

3.3.4.1 Experiment 1: Einfluss von Risikoaversion

Um den Einfluss der Risikoaversion auf den Verhandlungs-Anteil eines Agenten zu ermitteln, werden Simulationen in der *Divide-The-Grid*-Domäne durchgeführt. Die Feldgröße beträgt dabei 4×4, so dass es insgesamt 16 Zellen gibt, die von den Agenten beansprucht werden können. Die Startposition der Agenten ist dabei fixiert, so dass Agent 1 (blau in Abbildung 3.8) immer in der linken oberen Ecke startet und Agent 2 (rot in Abbildung 3.8) immer in der rechten unteren Ecke. Wenn die Verhandlung durch einen Agenten abgebrochen wird, indem eine bereits gefärbte Zelle des anderen Agenten beansprucht wird,

so führt das zu einer negativen Belohnung von $R_i = -1$ für beide Agenten $i \in \{1, 2\}$. Ist die Verhandlung erfolgreich, so erhält jeder Spieler i eine Belohnung proportional zu seinen beanspruchten Zellen, also $R_i = n_i/C$ wobei C die Gesamtanzahl an Zellen ist.



(a) Verhandlungsanteil Agent 1 (b) Verhandlungsanteil Agent 2

Abbildung 3.10: Das Verhandlungsergebnis wird durch die Risikoaversion der Agenten beeinflusst. Abgebildet sind die Verhandlungsanteile für Agent 1 (links) und Agent 2 (rechts), für unterschiedliche Niveaus der Risikoaversion, modelliert durch unterschiedliche Werte von α.

Wie im Abschnitt 3.3.3 beschrieben wird die Risikoaversion eines Agenten über den Konfidenzparameter α zur Berechnung des *CVaR*-Wertes gesteuert. Je kleiner α gewählt wird, desto geringer der Anteil der betrachteten Worst-Cases zur Berechnung von *CVaR*, woraus eine höhere Risikoaversion des Agenten resultiert. Zur Ermittlung des Einflusses der Risikoaversion des Agenten auf das Ergebnis, werden jeweils 10 unabhängige Trainingsläufe zu jeweils 8 verschiedenen Werten für α gemacht, wobei jeder Trainingslauf insgesamt über 120.000 Episoden umfasst. Aus der Kombination der 8 Werte für α, für beide Agenten ergeben sich 64 verschiedene Zusammensetzungen von Risiko-Präferenzen über beide Agenten.

Die Ergebnisse dieser 64 Konfigurationen sind in 3.3.4.1 dargestellt. Jede Zelle der dargestellten Heatmaps repräsentiert dabei den prozentualen Verhandlungsanteil eines Agenten, gemittelt über die 10 durchgeführten Simulationen. Abbildung 3.10a gibt dabei den prozentualen Verhandlungsanteil von Agent 1 an und Abbildung 3.10b den von Agent 2. Zu beachten ist, dass auf den Achsen $1-\alpha$ dargestellt ist, d.h. die Risikoaversion nimmt entlang der Achsen zu. Zur Ermittlung der Ergebnisse wurden jeweils die letzten 100 gespielten Episoden verwendet, um das Ergebnis repräsentativ für fertig trainierte Agenten zu halten. Abbildung 3.10a zeigt, dass eine Erhöhung der Risikoaversion von Agent 1, bei gleichzeitiger Verringerung der Risikoaversion von Agent 2, zu einer Verringerung des Verhandlungsergebnisses von Agent 1 führt (rechte

untere Ecke der Heatmap). Analog zeigt Abbildung 3.10b das gleiche Muster für eine Verringerung der Risikoaversion von Agent 2 und eine Erhöhung von Agent 1 (linke obere Ecke).

3.3.4.2 Experiment 2: Einfluss von Informationsasymmetrie

In diesem Experiment wird der Einfluss der Informationsasymmetrie auf den generellen Verhandlungserfolg zwischen den Agenten untersucht. Dafür wird die in Abschnitt 3.3.2.2 vorgestellte Käufer-Verkäufer-Domäne verwendet, wobei die Agenten durch zwei unabhängig trainierte Instanzen des Algorithmus *Categorical-DQN*, wie im vorigen Experiment, modelliert sind. Wie in Abbildung 3.7 dargestellt tritt Agent 1 (blau, auf der linken Seite) als Verkäufer und Agent 2 (rot, auf der rechten Seite) als Käufer auf. Beide Agenten müssen über den Verlauf einer Episode einen Preis festlegen, der die minimale Verkaufsbereitschaft (Verkäufer) bzw. die maximale Zahlungsbereitschaft (Käufer), in Form eines numerischen Preises angibt. Um Informationsasymmetrie einzuführen, tritt das Handelsgut in zwei Kategorien auf: das Handelsgut kann entweder geringe Qualität haben oder hohe Qualität, wobei lediglich der Verkäufer die tatsächliche Qualität des Gutes beobachten kann. Die Wahrscheinlichkeit, mit der das Handelsgut in einer der beiden Kategorien auftritt ist p_l für geringe Qualität und mit Gegenwahrscheinlichkeit ($p_h = 1 - p_l$) tritt es in hoher Qualität auf.

	Bewertung Verkäufer (U_s)	Bewertung Käufer (U_b)
Geringe Qualität	0,1	0,3
Hohe Qualität	0,4	0,6

Tabelle 3.1: Belohnungen für den Verkäufer und den Käufer, wenn das Handelsgut in hoher Qualität bzw. in geringer Qualität vorliegt.

Die Verhandlung ist erfolgreich, wenn am Ende einer Episode der Angebotspreis des Käufers p_b mindestens so hoch ist wie der Nachfragepreis des Verkäufers p_s. Der Verkäufer und der Käufer beziehen einen Nutzen U_s bzw. U_b aus dem Besitz des Handelsgutes, der, in Abhängigkeit der Qualität, in Tabelle 3.1 dargestellt ist. Ist die Verhandlung nicht erfolgreich, so entspricht die Belohnung des Verkäufers seinem Nutzen U_s aus dem Besitz des Handelsgutes, der Verkäufer bekommt in diesem Fall keine Belohnung. Bei erfolgreicher Verhandlung entspricht die Belohnung des Verkäufers dem erzielten Preis p abzüglich seines Nutzens $R_s = p - U_s$, die Belohnung des Käufers entspricht seinem Nutzen abzüglich des gezahlten Preises $R_b = U_b - p$. Für beide Qualitätsklassen besteht eine Differenz von $0,2$ zwischen den Nutzenniveaus der Agenten, so dass ein Verkauf sich für beide Seiten in jedem Fall lohnen würde, da sich beide Agenten durch einen Transfer besserstellen können.

Um nun den Effekt der Informationsasymmetrie in Form von Qualitätsunsicherheit auf den Verhandlungserfolg zu untersuchen, wird die Wahrschein-

lichkeit p_l für das Auftreten des Handelsgutes in geringer Qualität variiert. Da nur der Verkäufer die tatsächliche Qualität des Handelsgutes beobachten kann, entsteht dadurch Unsicherheit für den Käufer welche Qualität das Produkt tatsächlich hat. Für die Grenzfälle, also für $p_l = 0$ oder $p_l = 1$ gilt, dass das Handelsgut nur noch in einer Kategorie auftritt, d.h. in diesem Fall wäre es immer noch möglich dass der Käufer die richtige Qualität erlernt, obwohl er die Qualität nicht direkt beobachtet. Wenn das Handelsgut aber mit annähernd gleicher Wahrscheinlichkeit auftritt ($p_l = 0,5$), so besteht maximale Informationsasymmetrie zwischen den beiden Parteien.

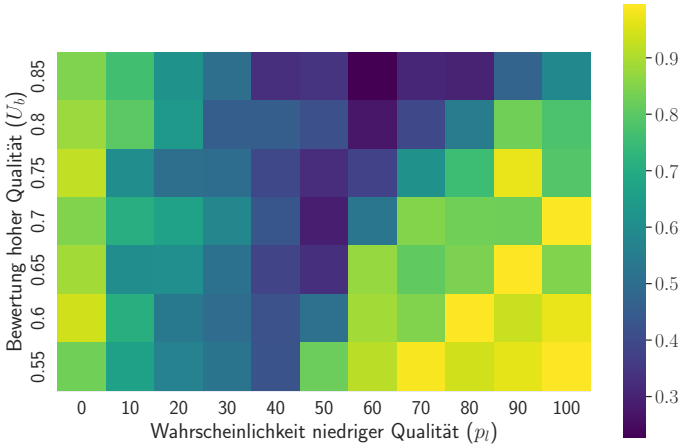


Abbildung 3.11: Verhandlungserfolg (Wahrscheinlichkeit) in der Verkäufer-Käufer-Domäne in Abhängigkeit des Auftretens des Produktes in niedriger Qualität p_l und der Bewertung des Gutes durch den Käufer (U_b).

Zur Auswertung dieses Experiments wird nun die Wahrscheinlichkeit für das Auftreten des Produkts in geringer Qualität p_l in Schritten von jeweils 10 Prozentpunkten erhöht. Für jeden Wert von p_l werden 10 unabhängige Läufe des gleichen Szenarios durchgeführt, um die Ergebnisse statistisch zu erhärten. Um auch den Effekt unterschiedlicher Bewertungen U_b für den Kauf des Handelsgutes in niedriger bzw. hoher Qualität zu untersuchen, wurden außerdem verschiedene Bewertungen für den Käufer getestet, so dass für die Bewertung des Käufers gilt $U_b \in \{0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85\}$. Aus der Kombination der 11 Wahrscheinlichkeitswerte für p_l und den 7 Belohnungen für den Käufer ergeben sich 77 Konfigurationen, die in Abbildung 3.11 aufgeschlüsselt sind. Jede Zelle gibt dabei die Wahrscheinlichkeit an, mit der am Ende einer Episode eine erfolgreiche Verhandlung zu Stande gekommen ist. Das Training umfasste insgesamt 120.000 Episoden, wobei jede Episode eine maximale Dauer von 11 Zeitschritten umfasst. Wie in Abbildung 3.11 zu sehen, zeigt sich insgesamt, dass der Erfolg der Verhandlung sowohl von der Wahrscheinlichkeit des Handelsgutes in einer Kategorie abhängt als auch von der Bewertung, die der Käufer durch den erfolgreichen Erwerb des Gutes in hoher Qualität

erhält. Wenn die Bewertung des Käufers für das Handelsgut in hoher Qualität zunimmt, so steigt auch die Erfolgsquote, allerdings tritt dieser Fall vermehrt auf bei einem höheren Anteil an hochwertigen Artikeln.

3.3.5 Zusammenfassung und Diskussion

In diesem Abschnitt wurden Verhandlungsdomänen für bilaterale Verhandlungen mittels deep Reinforcement Learning beschrieben. Dazu wurde ein Überblick über verwandte Arbeiten aus dem Bereich der Spieltheorie bzw. der evolutionären Spieltheorie gegeben. Die beiden vorgestellten Verhandlungsdomänen modellieren dabei unterschiedliche Aspekte eines Verhandlungsprozesses zwischen zwei Agenten. Die erste Domäne modelliert eine Aufteilungsverhandlung zwischen zwei Akteuren, die zweite Domäne stellt eine Verkaufssituation dar, so dass die Agenten unterschiedliche Rollen einnehmen. Um den Einfluss der Risikoaversion der Verhandlungspartner auf ihren Anteil abzuschätzen, wurde ebenfalls ein Agenten-Modellierung vorgestellt, die das Treffen risikoaverser Entscheidungen ermöglicht, basierend auf auf dem Algorithmus *Categorical DQN* [12]. Diese Agentenarchitektur ermöglicht die Justierung der Risikoaversion eines Agenten durch das Setzen eines einzelnen Parameters.

In dem ersten Experiment basierend auf der Aufteilungs-Verhandlungsdomäne konnte gezeigt werden, dass sich die Risikoaversion eines Teilnehmers negativ auf seinen Verhandlungsanteil auswirkt. Dieses Ergebnis ist im Einklang zu spieltheoretischen Resultaten [72]. Das dies auch im Kontext lernender Agenten gilt, kann als empirischer Nachweis der Gültigkeit axiomatischer spieltheoretischer Modelle für KI-Systeme verstanden werden. Eine interessante Stoßrichtung für weitere Experimente wäre es dabei, Agenten mit unterschiedlichen Risikometriken gegeneinander antreten zu lassen. Darüber hinaus wäre es wichtig, andere mögliche Einflüsse auf das Verhandlungsergebnis zu analysieren, die beispielsweise aus menschlichen Verhandlungen bekannt sind, wie Sturheit oder die Fähigkeit zu bluffen.

Um den Einfluss von Informationsasymmetrie auf den Verhandlungserfolg zu untersuchen, wurde die Käufer-Verkäufer-Domäne entwickelt, wodurch die Informationen, die die beiden Verhandlungspartner erhalten, unterschiedlich gewichtet werden können. Informationsasymmetrie tritt hier in Form von Qualitätsunsicherheit auf, d.h. ein Produkt kann verschiedenen Qualitätsklassen angehören. Die Informationsasymmetrie wurde in ökonomischen Veröffentlichungen bereits als kritischer Faktor für das Funktionieren eines Marktes beschrieben [3]. Wiederum sind zwei Akteure am Verhandlungsprozess beteiligt, wobei Akteur 1 der Verkäufer und Akteur 2 der Käufer eines Artikels ist. Der Verkäufer hat in diesem Spiel Informationen über die Qualitätsklasse des aktuellen Produkts, während der Käufer keine solchen Informationen erhält. In dem beschriebenen Experiment konnte gezeigt werden, dass sich asymmetrisch verteilte Informationen zwischen den Teilnehmern negativ auf den Verhandlungserfolg auswirken. Auch hier dient das Experiment als empirischer Nachweis

dafür, dass theoretisch fundierte Modelle für Reinforcement Learning Gültigkeit haben können.

Bei den vorgestellten Verhandlungsmodellierungen zwischen zwei Agenten wird von bestimmten Aspekten echter Verhandlungssituationen abstrahiert. Wesentliche Unterschiede der in dieser Arbeit vorgestellten Modellierung, im Vergleich zu Verhandlungen zwischen beispielsweise Menschen, sind das Vorhandensein bereits gewonnener Verhandlungserfahrungen sowie die Fähigkeit den Verhandlungspartner einzuschätzen (Gegnermodellierung).

3.3.5.0.1 Vorerfahrungen In dem hier vorgestellten Ansatz wird davon ausgegangen, dass beide Akteure ohne weiteres Vorwissen in den Trainingsprozess einsteigen. Diese Herangehensweise vereinfacht das Training erheblich, da beide Agenten als Instanzen des gleichen Lernalgorithmus geführt werden. In echten Anwendungen ist davon auszugehen, dass eine KI-Komponente kontinuierlich neue Erfahrungen sammeln muss und immer wieder mit neuen Verhandlungspartnern interagiert, so dass ein breites Spektrum an Situationen im Erfahrungsschatz abgebildet wird. Die vorgestellten Verhandlungsdomänen bieten jedoch die Möglichkeit, auch unterschiedliche Agentenarchitekturen zu kombinieren, da die Schnittstelle lediglich das Setzen einer Aktion erfordert.

3.3.5.0.2 Gegnermodellierung Bei vielen Verhandlungssituationen kann davon ausgegangen werden, dass die Informationen zwischen den Verhandlungspartnern nicht im gleichen Maße verfügbar sind. Dies betrifft nicht nur den potentiellen Handelsgegenstand (sowie beispielsweise die Qualität des gehandelten Gutes), sondern auch Informationen über den Verhandlungspartner selbst. Hat einer der Teilnehmer Informationen über die Präferenzen oder Wünsche des Gegenübers, so wäre es für ihn einfacher, ein darauf abgestimmtes Angebot abzugeben, so dass eventuell ein anderes Ergebnis resultiert, als in einer völlig unvoreingenommenen Ausgangslage.

Das Erstellen von Modellen anderer Agenten wird als Gegner-Modellierung (engl. *Opponent Modelling*) bezeichnet. In Bezug auf Verhandlungssituationen sind die drei Hauptaspekte der Modellierung: Präferenzschätzung (was will der Gegner?), Strategievorhersage (was wird der Gegner tun?) und Klassifizierung des Gegners (welcher Spielertyp ist der Gegner?) [8]. Ansätze zur Erstellung von Gegnermodellen für Verhandlungen in Multiagenten-Umgebungen stammen aus verschiedenen Bereichen, darunter Bayes'sches Lernen [30], nichtlineare Regression [31], genetische Algorithmen [65] und künstliche neuronale Netze [24]. Im Gegensatz zu Ansätzen aus dem Bereich der Gegner-Modellierung wird in dieser Arbeit nicht explizit ein Modell des Gegners erstellt. Vielmehr betrachtet ein Agent andere Agenten als Teil der Umgebung, was das Lernproblem aus Sicht eines einzelnen Agenten nicht-stationär macht. In diesem Fall ergibt sich die Anpassung an das gegnerische Verhalten aus der sich ändernden Datenverteilung, die ein Agent in seinem sequentiellen Speicher hat und die er zum Trainieren seiner Strategie verwendet.

4 Konditionale und bedingungslose Marktmechanismen

In diesem Kapitel werden nun zwei konkrete Marktmechanismen vorgestellt. Der erste Abschnitt beschreibt einen konditionalen Mechanismus, bei dem die Bedingungen für einen Transfer an die ausgeführten Aktionen der Handelspartner gebunden sind. Der zweite Mechanismus ist bedingungslos, so dass die Agenten ihre Belohnungen lose koppeln können, ohne das dadurch bestimmte Anforderungen an deren Verhalten gebunden sind. Der konditionale Marktmechanismus wird zuerst im Kontext von Normalformspielen mit zwei Spielern beschrieben und anschließend auch in komplexeren sequentiellen Entscheidungsproblemen evaluiert. Ein Vergleich der beiden Mechanismen wird in einer Simulation eines Fabrik-Szenarios vorgenommen, wobei bis zu 16 unabhängige Agenten involviert sind. Die Marktmechanismen werden mit unterschiedlichen Algorithmen kombiniert, um so einen Vergleich über verschiedene Verfahren zu ermöglichen.

4.1 Vorveröffentlichungen

Die Kerninhalte dieses Kapitels wurden vom Autor bereits in [74] bzw. in [76] publiziert. Der Ansatz zum Handeln von Aktionen in Abschnitt 4.2 wurde vom Autor bereits in [74] veröffentlicht und der bedingungslose Marktmechanismus wurde vom Autor bereits in [76] veröffentlicht. Die in diesem Kapitel dargestellten Abbildungen sind der Art nach ebenfalls bereits in [74] bzw. in [76] veröffentlicht, wurden im Rahmen der Ausarbeitung dieser Arbeit aber neu visualisiert und beschriftet. Die Datengrundlage für die Abbildungen hat sich dabei jedoch nicht geändert und entspricht den gleichen Auswertungen wie in [74] bzw. in [76] vorgestellt.

4.2 Konditionaler Markt: Action Trading

Im Folgenden wird ein konditionaler Marktmechanismus zum Handeln von Aktionen zwischen den Agenten beschrieben, der als *Action Trading* bezeichnet wird. Das Konzept wird zuerst anhand eines Normalformspiels mit zwei Spielern erläutert und anschließend in sequentiellen Entscheidungsproblemen mit

mehr als zwei Agenten evaluiert.

4.2.1 Idee und Motivation

Die Idee für den bedingten Marktmechanismus zum Handeln von Aktionen, beruht auf der Beobachtung, dass ein System von unabhängig lernenden Agenten wenig Chance auf das Erlernen kooperativer Verhaltensweisen hat, wenn sich die Agenten nicht explizit gegenseitig für bestimmte Verhaltensweisen belohnen dürfen. Die in diesem Kapitel vorgestellte Marktform wird daher auch *Action Trading* genannt, da Agenten anderen Agenten für deren ausgeführte Aktionen Belohnungen zukommen lassen können. Dadurch ermöglicht es den Agenten andere Agenten zu bestimmten Verhaltensweisen zu incentivieren, so dass mögliche Interessenskonflikte aufgelöst werden können. Erleidet ein Agent i durch das Verhalten eines anderen Agenten j einen Schaden bzw. eine geringere Belohnung als unter einem anderen Verhalten von j, so wäre eine mögliche Lösung, dass Agent i Agent j zu einem anderen Verhalten motiviert. Da es sich bei beiden Agenten um nutzenmaximierende Entscheidungsträger handelt, wird sich nur dann eine Verhaltensänderung bewirken lassen, wenn j für den Schaden, der ihm durch die Verhaltensänderung entsteht, entlohnt wird. Im Idealfall entsteht Agent i durch die Verhaltensänderung von j ein so hoher Gewinn, dass es möglich ist, j mindestens für seinen Schaden zu kompensieren oder sogar überzukompensieren, so dass sich beide Agenten dadurch besserstellen als im Vergleich zu der Ausgangslage.

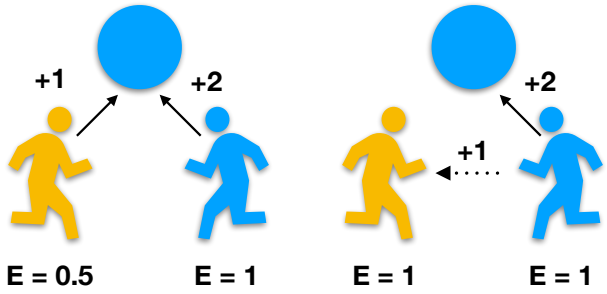


Abbildung 4.1: Das Coin Game: Zwei Agenten konkurrieren um eine Münze. Während rein eigennütziges Verhalten ohne Handel die Agenten zu gierigem Handeln anregt, kann die Einführung eines Marktes dazu beitragen, den erwarteten Wert beider Agenten zu erhöhen.

Abbildung 4.1 veranschaulicht, wie die Einführung eines konditionalen Marktes dazu beitragen kann, Interessenskonflikte zwischen zwei konkurrierenden Agenten aufzulösen. Angenommen zwei Agenten (gelb und blau) konkurrieren um eine Münze, wobei der gelbe Agent eine Belohnung von +1 für das Einsammeln der Münze erhält, während der blaue Agent eine Belohnung von +2 dafür bekommt. Es sei weiter angenommen, dass für beide Agenten

eine Wahrscheinlichkeit von $0,5$ besteht, die Münze zu bekommen, wenn beide es gleichermaßen versuchen. Würden die Agenten das Spiel unendlich oft gegeneinander spielen, so würde sich langfristig eine erwartete Belohnung von $0,5$ für den gelben Agenten und eine erwartete Belohnung von 1 für den blauen Agenten ergeben. Da jeder Agent nur seine eigene Belohnung in Betracht zieht, besteht für den gelben Agenten kein Anreiz, dem blauen Agenten die Münze zu überlassen, obwohl dies das Gesamtergebnis maximieren würde. Dies ändert sich, wenn die Agenten in die Lage versetzt werden Belohnungen gegen bestimmte Aktionen des Gegenübers auszutauschen. Wenn sie tauschen können, könnte der blaue Agent dem gelben Agenten eine Belohnung von $+1$ vorschlagen, wenn der gelbe Agent die Münze nicht nimmt. In diesem Fall sind die Belohnungen beider Agenten $+1$, was für beide eine Verbesserung darstellt.

Das Action Trading definiert also einen Kaufvertrag zwischen einer Verkäufer- und einer Käuferseite, ähnlich wie in einer realen Kaufsituation, wo sich zwei Parteien auf eine Zahlung durch den Abschluss eines Kaufvertrags einigen. Der Mechanismus ist konditional, da der Handel auf einer Leistung und einer Gegenleistung beruht, es findet also ein expliziter Austausch von Ressourcen oder Leistungen zwischen den Agenten statt. Da jeder Transfer immer auf der Zustimmung beider Seiten beruht, wird durch den Handel stets eine Pareto-Verbesserung erreicht, da sich beide Agenten dadurch echt besserstellen (zumindest unter der Annahme, dass beide Seiten bereits eine optimale Policy erlernt haben). Durch diese Pareto-Verbesserungen können sich also sowohl die Gesamtbelohnung als auch die individuell erzielten Belohnungen verbessern.

4.2.2 Verwandte Arbeiten

Verwandte Arbeiten zu dem hier vorgestellten Ansatz Action Trading kommen insbesondere aus dem Bereich des Multiagent Reinforcement Learning. Im Folgenden werden Ansätze vorgestellt, bei denen die Modifikation des Belohnungssystems nicht direkt durch die Agenten erfolgt, sondern exogen vorgenommen wird sowie Methoden, bei denen die Agenten direkt die Belohnungen anderer Agenten modifizieren können, d.h., einen Peer-to-Peer Austausch ermöglichen.

4.2.2.0.1 Modifikation des Belohnungssystems Die Möglichkeit zur Steuerung des Kooperationsgrades der Agenten über das Belohnungssystem zeigen Tampuu et al. in [86]. Die Autoren modifizieren dazu die Belohnungsfunktion des Atari Videospiels Pong, um die beiden Agenten dazu zu bringen entweder zusammenzuarbeiten oder kompetitiv zu spielen. Es wird gezeigt, dass sich durch die Wahl des Belohnungssystems rein kooperative bzw. nichtkooperative Policies erzeugen lassen. Ein weiterer Ansatz, der auf der Modifikation des Belohnungssystem beruht, wird in [67] vorgestellt. Dabei werden die Belohnungen von Agenten so modelliert, dass darin ebenfalls zu einem bestimmten Grad die Belohnungen anderer Agenten berücksichtigt werden. Durch dieses Belohnungsschema bilden sich Gruppen autonomer Agenten, die

mit höherer Wahrscheinlichkeit zu sozial verträglichen Ergebnissen konvergieren, als wenn nur die individuellen Belohnungen betrachtet werden. Der in diesem Abschnitt vorgestellte Ansatz zum Handeln von Aktionen unterscheidet sich in der Funktionsweise zu den obigen Arbeiten dadurch, dass das Belohnungssystem nicht von außen verändert wird. Vielmehr können die Agenten durch ihren erweiterten Aktionsraum das Belohnungssystem selbständig verändern, so dass sich andere Verhaltensweisen emergent ergeben.

4.2.2.0.2 Peer-to-Peer Austauschmechanismen Ein Ansatz bei dem Agenten lernen sich direkt zu incentivieren wird in [96] vorgestellt. Dabei lernen die Agenten zwei unterschiedliche Funktionen: zum einen lernen sie ihre gewöhnliche Policy, die darüber entscheidet welche Aktion zu einem Zeitpunkt ausgeführt werden soll. Zusätzlich wird eine Incentive-Funktion erlernt, die darüber entscheidet, wie viel Belohnung anderen Agenten für bestimmte Aktionen gegeben werden soll. Für den Fall des Gefangen-Dilemmas in Kombination mit einer Policy-Gradienten basierten Lernregel konnte die Konvergenz dieses Ansatzes zu dem sozialen Optimum beidseitiger Kooperation nachgewiesen werden. Ein wichtiger Unterschied zu dem in diesem Abschnitt vorgestellten Ansatz besteht darin, dass in [96] zur Incentivierung anderer Agenten Belohnungen neu erzeugt werden können, d.h. dass die Agenten mehr Belohnung vergeben können als sie selbst von der Umgebung erhalten. In dem hier vorgestellten Ansatz Action Trading können Belohnungen nur vergeben werden, die auch über die Umgebung erwirtschaftet werden. Eine Erzeugung neuer Belohnungen ist nicht möglich, da dies insbesondere für reale Anwendungen nur schwer gerechtfertigt werden kann.

Ein weiterer Ansatz bei dem Agenten Belohnungen direkt vergeben können ist in [49] beschrieben. Der Ansatz beruht auf der Idee, dass Agenten Belohnungen übertragen können, um sich gegenseitig zu kooperativen Verhaltensweisen zu motivieren. Der Ansatz unterscheidet sich von dem Ansatz in diesem Kapitel insbesondere dadurch, dass es keinen bidirektionalen Austausch zwischen Agenten gibt, sondern der Transfer nur von einem Agenten zu einem anderen geht, ohne eine Gegenleistung, daher auch der dafür gewählte Name *Gifting*. Der Vorgang stellt also weniger einen Handelsmechanismus dar, als mehr ein Geschenk von einem Agenten an einen anderen.

4.2.3 Konzept

In diesem Abschnitt wird der Algorithmus Action Trading vorgestellt. Der Algorithmus realisiert den bidirektionalen Austausch von Aktionen gegen Belohnungen in Form von Transaktionen. Eine Transaktion definiert ein Tupel $t = (a_t, r_t)$ bestehend aus einer zu handelnden Aktion a_t zum Zeitpunkt t, sowie einer Belohnung r_t. Die Aktion $a_t \in \mathcal{A}$ definiert daher die Ressource des Handelspartners, die gegen die Belohnung r_t des anderen Handelspartners, getauscht wird. Formal lässt sich Action Trading durch eine Erweiterung des

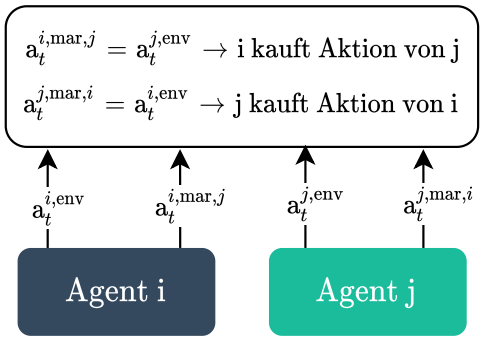


Abbildung 4.2: Der Ansatz Action Trading lässt Agenten Belohnungen gegen bestimmte Aktionen tauschen. Agenten wählen also zusätzlich zu ihren ursprünglichen Aktionen auch Marktaktionen. Ein Handel kommt zustande, wenn ein Angebot für eine Aktion mit einer tatsächlich ausgeführten Aktion übereinstimmt.

stochastischen Spiels $\mathcal{G} = (\mathcal{N}, \mathcal{A}^1, ..., \mathcal{A}^{\mathcal{N}}, \mathcal{S}, \mathcal{T}, \mathcal{R}^1, ..., \mathcal{R}^{\mathcal{N}})$ definieren, bestehend aus einer Menge von Agenten $\mathcal{N}$, einer Menge von Zuständen $\mathcal{S}$, einer Menge von Aktionen $\mathcal{A}^i$, für jeden Agenten $i \in \mathcal{N}$, der Zustandstransitionsfunktion $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \rightarrow PD(\mathcal{S})$ und einer Belohnungsfunktion $R^i : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \rightarrow \mathbb{R}$ für jeden Agenten $i \in \mathcal{N}$. Action Trading beruht auf drei Kern-Erweiterungen des stochastischen Spiels:

- Jeder Agent i erhält zusätzlich zu seinem originalen Aktionsraum $\mathcal{A}^i$ weitere Marktaktionen $\mathcal{A}^i_m$ durch die mit anderen Agenten gehandelt werden kann.
- Die Einführung einer Marktfunktion $\mathcal{M} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \times \mathcal{A}^1_m \times ... \times \mathcal{A}^N_m \times \mathbb{R}^N \rightarrow \mathbb{R}^N$, die die Umverteilung der Belohnungen gemäß der ausgeführten Transaktionen berechnet.
- $P : \mathcal{S} \rightarrow \mathbb{R}$ eine (zustandsabhängige) Preisfunktion, die die Höhe der Belohnung im Austausch für eine Aktion definiert.

Das Handeln mit Aktionen wird ermöglicht, indem Agenten über die erweiterten Marktaktionen $\mathcal{A}^m_i$ Angebote für bestimmte Aktionen anderer Agenten abgeben können, wodurch ein Markt für Aktionen geschaffen wird (auch Aktionsmarkt genannt). Jeder Agent $i \in \mathcal{N}$ wählt in dieser erweiterten Form des stochastischen Spiels zu jedem Schritt t eine Aktion $\mathbf{a}^i_t = (a^{i,\text{env}}_t, a^{i,\text{mar},j}_t)$, die zwei Komponenten beinhaltet: $a^{i,\text{env}}_t \in \mathcal{A}^i$ ist die Aktion, die Agent i zum Zeitpunkt t ausführt und $a^{i,\text{mar},j}_t \in \mathcal{A}^i_m$ ist die Aktion, die Agent i für Agent j zur Ausführung zu Zeitpunkt t vorschlägt. Führt Agent j die vorgeschlagene Aktion zum Zeitpunkt t aus, das heißt wenn $a^{j,\text{env}}_t$ mit $a^{i,\text{mar},j}_t$ übereinstimmt, dann ist die Bedingung für die Transaktion der Belohnung erfüllt und Agent i muss den, durch die Preisfunktion P definierten Preis, an j entrichten. Der Ablauf für einen Zeitschritt ist in Abbildung 4.2 dargestellt.

Die Marktfunktion ermittelt die aus dem Handel resultierenden Belohnungen pro Zeitschritt. Die Belohnungen, die aus einem Handel zwischen i und j resultieren werden durch die Marktfunktion folgendermaßen berechnet:

$$r_i = r_i - p * \delta_{a_t^{j,\text{env}}, a_t^{i,\text{mar},j}}$$

$$r_j = r_j - p * \delta_{a_t^{i,\text{env}}, a_t^{j,\text{mar},i}}$$

wobei $\delta_{x,y}$ für das Kronecker-Delta steht, mit $\delta_{x,y} = 1$ wenn $x = y$ und $\delta_{x,y} = 0$ falls $x \neq y$. Zur Berechnung der Belohnungen mit Action Trading kann die Marktfunktion als Prozedur, so wie in Algorithmus 3 in Form von Pseudo-Code dargestellt, verwendet werden. Dabei werden alle Transaktionen in einer Matrix $B \in \mathbb{N}^{|\mathcal{N}|\times|\mathcal{N}|}$ (der Bilanz) gespeichert. Die Kaufangebote aller Agenten werden mit den tatsächlich ausgeführten Aktionen der Agenten verglichen. Bei Übereinstimmung wird eine erfolgreiche Transaktion mit Transaktionswert p in der Bilanz vermerkt. Anschließend erfolgt die Neuberechnung der Belohnungen basierend auf der aktuellen Bilanz und den Belohnungen zum Zeitpunkt t.

Algorithm 3 Marktfunktion: Action Trading

```
1: procedure ActionTrading(a_t, r_t, s_t)
2:     B ← s_t                                   ▷ Aktueller Stand der Bilanz
3:     for agent i in N do
4:         for agent j in N \ i do
5:             a_t^{i,env}, a_t^{j,mar,i} ← a_t
6:             if a_t^{i,env} == a_t^{j,mar,i} then
7:                 B_{i,j} ← p                   ▷ j kauft Aktion von i
8:     r_t' ← b, r_t                             ▷ Berechne Belohnungen
9:     return r_t'
```

4.2.4 Action Trading für Normalformspiele mit zwei Spielern

Im Folgenden wird nun die Methode Action Trading für ein Spiel mit zwei Spielern und zwei Aktionen in Normalform dargestellt. Das betrachtete Spiel (dargestellt in Grafik 4.3) enthält zwei Nash-Gleichgewichte in reinen Strategien, gekennzeichnet durch (a_1, a_1) und (a_1, a_2), da ausgehend von diesen Aktionen kein Spieler einen Anreiz hat eine andere Aktion zu wählen, solange der andere das ebenfalls nicht tut. Das Gleichgewicht an der Stelle (a_1, a_1) ist suboptimal, da die Gesamtbelohnung im Falle von (a_1, a_2) größer ist.

Dieses Spiel lässt sich nun durch folgende Erweiterungen in ein Spiel mit mit Action Trading überführen. Die Aktionsräume der beiden Spieler $\mathcal{A}^1$ und $\mathcal{A}^2$ mit $\mathcal{A}_i = \{a_1, a_2\}$ für $i \in \{1, 2\}$ werden erweitert, so dass $\mathcal{A}^1_{\text{AT}} = \mathcal{A}^2_{\text{AT}} = \{(a_1, \text{no-op}), (a_1, a_1), (a_1, a_2), (a_2, \text{no-op}), (a_2, a_1), (a_2, a_2)\}$, wobei für ein Tupel $(x, y) \in \mathcal{A}^i_{\text{AT}}$ gilt, dass x die auszuführende Aktion des Spielers definiert und y

	a_1	a_2
a_1	$0.5, 0.0$	$4.0, 0.0$
a_2	$0.0, 1.0$	$1.0, 0.0$

Abbildung 4.3: Spiel mit 2 Nash-Gleichgewichten.

die Handelsaktion, also die Aktion, für die dem anderen Spieler ein Handel angeboten wird. Die Aktion no-op kennzeichnet dabei den Fall, dass kein Handel vorgeschlagen werden soll, so dass das Handeln nicht obligatorisch ist. Das auf diese Art erweiterte Spiel ist in Abbildung 4.4 dargestellt. Der Preis p ist dabei ein frei wählbarer Parameter und bestimmt die Höhe des Belohnungstransfers zwischen den Agenten.

	$(a_1, \text{no-op})$	(a_1, a_1)	(a_1, a_2)	$(a_2, \text{no-op})$	(a_2, a_1)	(a_2, a_2)
$(a_1, \text{no-op})$	$0.5, 0.0$	$4.0, 0.0$	$0.5, 0.0$	$4.0, 0.0$	$4.0 + p, \ -p$	$4.0, 0.0$
(a_1, a_1)	$0.0, 1.0$	$1.0, 0.0$	$0.5 - p, p$	$4.0, 00$	$4.0 + p, \ -p$	$4.0, 0.0$
(a_1, a_2)	$0.5, 0.0$	$0.5 + p, \ -p$	$0.5, 0.0$	$4.0, 00$	$4.0, 0.0$	$4.0 - p, p$
$(a_2, \text{no-op})$	$0.0, 1.0$	$0.0, 1.0$	$p, 1.0 - p$	$1.0, 0.0$	$1.0, 0.0$	$1.0 + p, \ -p$
(a_2, a_1)	$-p, 1.0 + p$	$-p, 1.0 + p$	$0.0, 1.0$	$1.0, 0.0$	$1.0, 0.0$	$1.0 + p, \ -p$
(a_2, a_2)	$0.0, 1.0$	$0.0, 1.0$	$p, 1.0 - p$	$1.0 - p, p$	$1.0 - p, p$	$1.0, 0.0$

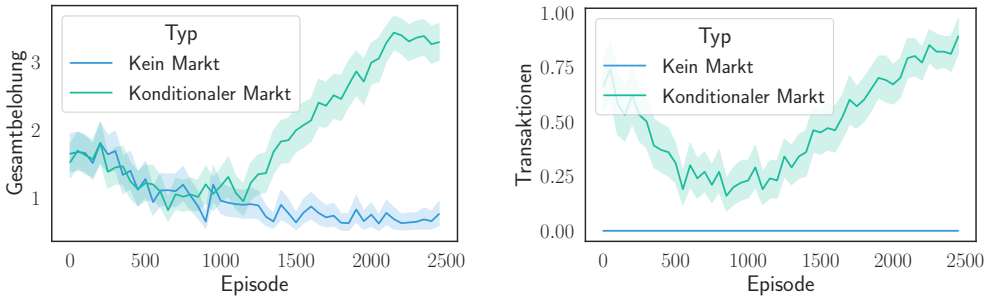
Abbildung 4.4: Das erweiterte Spiel mit Action Trading.

4.2.4.1 Evaluation

Um herauszufinden, wie sich die Erweiterung durch Action Trading auf das Spiel bezüglich der Erlernbarkeit des globalen Optimums auswirkt, werden nun zwei Agenten mit Methoden des Reinforcement Learning trainiert. Dazu wird der Q-Learning-Algorithmus in tabularer Form verwendet, wobei jeder Agent als eigenständige Instanz des Algorithmus modelliert wird. Die Aktualisierung der Q-Funktion erfolgt dabei anhand der Regel:

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha\big[r_i + \gamma \max_{a' \in \mathcal{A}^i} Q_i(s',a') - Q_i(s,a)\big]$$

Für das Q-Learning wird dabei eine Lernrate von $\alpha = 0,001$ und eine Diskontierung von $\gamma = 1,0$ verwendet. Für das Action Trading wird ein Preis von $p = 1$ gesetzt. Alle Experimente umfassen 2500 Trainingsschritte. Um kontinuierliche Exploration während des Trainings zu gewährleisten, wird das *Epsilon-*



(a) Gesamtbelohnung

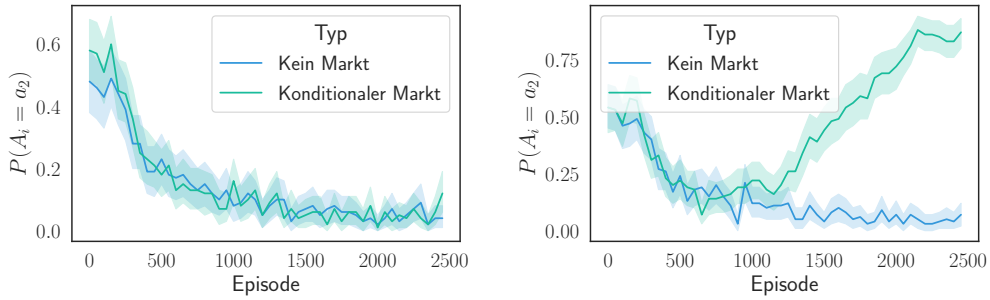
(b) Transaktionen

Abbildung 4.5: Gesamtbelohnung und Anzahl der Transaktionen des Matrix-Spiels aus Abbildung 4.3.

Greedy-Verfahren verwendet, so dass eine zufällige Aktion mit Wahrscheinlichkeit ϵ gewählt wird und eine optimale Aktion gemäß aktueller Q-Funktion mit Wahrscheinlichkeit $1 - \epsilon$. Dabei wird ϵ linear über den Trainingsverlauf bis zu einem minimalen Wert von $\epsilon = 0,1$ abgekühlt. Um die Ergebnisse statistisch zu erhärten, werden für jedes Spiel 100 unabhängige Trainingsläufe ausgeführt und die Ergebnisse gemittelt.

Die erzielte Gesamtbelohnung und die gespielten Aktionen sind in Abbildung 4.5 dargestellt. Zu Beginn des Trainings ist die Gesamtbelohnung für beide Verfahren, bedingt durch die stark zufällig geprägte Aktionswahl der Agenten auf einem geringen Level. Im Falle des Trainings ohne Marktmechanismus fällt die Gesamtbelohnung über den weiteren Verlauf jedoch weiter ab und konvergiert langsam in Richtung des suboptimalen Nash-Gleichgewichts (a_1, a_1), das eine Gesamtbelohnung von $0,5$ bringt. Im Vergleich dazu lassen sich die Lerndynamiken durch die Integration das Action Trading positiv beeinflussen. Das lässt sich an der steigenden Gesamtbelohnung (grüne Kurve) erkennen, die sich ungefähr bei der Hälfte der Trainingsschritte in Richtung des globalen Optimums bewegt. Auch hier wird über die 100 gemittelten Durchläufe nicht immer das Optimum erreicht, aber durchschnittlich ist eine signifikante Verbesserung zu beobachten.

Die höhere erzielte Gesamtbelohnung, die durch das Action Trading erreicht wird, spiegelt sich in den durchgeführten Transaktionen wider. Es zeigt sich, das gegen Ende des Training in den meisten Schritten eine Transaktion zwischen den Agenten durchgeführt wird. Das höhere Gesamtergebnis im Falle des konditionalen Marktes wird erzielt, da im Gegensatz zu dem Fall ohne Markt Agent 2 nicht mehr die global suboptimale Aktion a_1 spielt, sondern sich mit höherer Wahrscheinlichkeit für Aktion a_2 entscheidet. Abbildung 4.6 zeigt die Wahrscheinlichkeiten beider Agenten Aktion a_2 zu spielen, also $P(A_i = a_2)$ wobei gilt, dass $P(A_i = a_1) = 1 - P(A_i = a_2)$. Während Agent 1 weiterhin Aktion a_1 wählt, lernt Agent 2, dass es sich lohnt Aktion a_2 zu wählen für den Fall mit Action Trading.



(a) Aktionswahl Agent 1

(b) Aktionswahl Agent 2

Abbildung 4.6: Aktionswahl mit und ohne Action Trading in einem zwei-Spieler-Spiel.

4.2.5 Action Trading in sequentiellen sozialen Dilemmas

Nachdem im vorigen Abschnitt Action Trading für den Fall eines Spiels in Normalform mit zwei Spielern erläutert wurde, wird in diesem Abschnitt die Anwendung in einem zeitlich erweiterten, d.h. sequentiellen Entscheidungsproblem mit bis zu vier Agenten, beschrieben. Das Szenario ist bekannt unter dem Namen Coin Game und wurde zuerst vorgestellt von Lerer und Peysakhovich in [46]. Das Coin Game umfasst in der ursprünglich vorgestellten Variante zwei Agenten, deren Ziel es ist, Münzen in einer zweidimensionalen Grid-Welt einzusammeln. Es gibt zwei verschiedene Typen von Münzen. Jeder Münzentyp gehört zu einem Agenten, wobei die Zugehörigkeit über die Farbe von Münze und Agent gekennzeichnet ist. Die Agenten haben das Ziel so viele Münzen wie möglich zu sammeln. Jedoch gilt, dass wenn Agent i eine Münze von Agent j einsammelt, so erhält j eine Bestrafung in Form einer negativen Belohnung. Für den Agenten, der die Münze einsammelt, spielt es jedoch keine Rolle, welche Münze gesammelt wird und er erhält in jedem Fall eine positive Belohnung.

Das Dilemma des Coin Games liegt also darin, dass ein Agent durch das Sammeln der (falschen) Münzen, dem anderen Agenten Schaden zufügen kann, er aber selbst keinerlei Kosten für diesen verursachten Schaden tragen muss. Aus Sicht eines Agenten ist dieser zugefügte Schaden also *extern*, da er nicht Teil des eigenen zu optimierenden Belohnungssystems ist. Lerer und Peysakhovich stellen in der Veröffentlichung zu der Domäne einen Ansatz vor, der auf der spieltheoretischen Strategie *Tit-for-Tat* beruht, die durch Methoden des *deep* Reinforcement Learning gelernt wird. Abbildung 4.7 zeigt eine Situation, bei der der grüne Agent und der blaue Agent sich auf die blaue Münze zubewegen. Da der grüne Agent bereits näher ist würde er im nächsten Schritt die Münze erhalten, was ihm eine Belohnung von $+1$ bringt, wohingegen der blaue Agent eine Bestrafung von -2 erhält, da es sich um eine blaue Münze gehandelt hat. Das Coin Game kann als ein sequentielles soziales Dilemma verstanden werden (vgl. Abschnitt 2.1.4), da es verschiedene Aspekte realer Dilemmas, wie

zeitliche Erweiterung oder Abstufungen kooperativen Verhaltens zwischen den Agenten berücksichtigt.

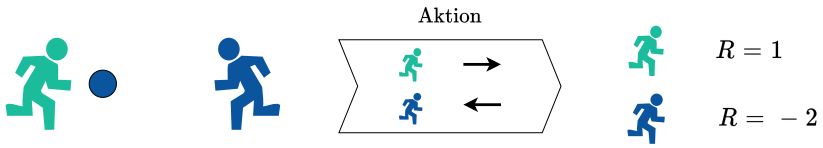


Abbildung 4.7: Das Coin Game: zwei Agenten konkurrieren um Münzen. Sammelt ein Agent die Münze des anderen, erhält dieser eine negative Belohnung.

Abbildung 4.8 zeigt, wie sich das Dilemma des Coin Games durch die Integration eines konditionalen Marktes auflösen lässt. Die Ausgangslage ist wie in Abbildung 4.7, jedoch gibt es jetzt die Möglichkeit für beide Agenten dem jeweils anderen einen Handel für eine bestimmte Aktion des Gegenübers vorzuschlagen. In diesem Fall kann der blaue Agent dem grünen Agenten vorschlagen die Richtung zu wechseln. Wenn der grüne Agent diesem Handelsangebot nachkommt und es dem blauen Agenten dadurch möglich wird die blaue Münze einzusammeln, ergibt sich für den grünen Agenten eine Belohnung von $+1$ (bei einem gesetzten Preis von $p = 1$) und für den blauen Agenten eine Belohnung von $1 - 1 = 0$, d.h. beide Agenten können sich durch den Handel echt besserstellen im Vergleich zu der Situation davor.

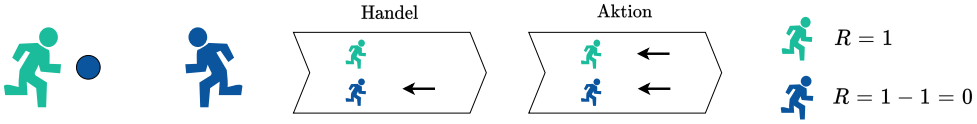


Abbildung 4.8: Durch Action Trading lässt sich das Dilemma im Coin Game auflösen.

4.2.5.1 Umsetzung im Coin Game

Zur Realisierung von Action Trading im Coin Game werden die Aktionsräume $\mathcal{A}^i$ jedes Agenten $i \in \mathcal{N}$ um die Handelsaktionen erweitert. Das Coin Game sieht vier mögliche Aktionen zur Navigation innerhalb des Felds vor, d.h. $\mathcal{A}^i = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$. Dieser diskrete Aktionsraum $\mathcal{A}^i$ wird durch die Hinzunahme der Handelsaktionen $\mathcal{A}^i_{\text{mar}} = \{\leftarrow, \rightarrow, \uparrow, \downarrow, \text{no-op}\}$ zu einem Multi-diskreten Aktionsraum $\mathcal{A}^i_{\text{SMS}} = \mathcal{A}^i \times \mathcal{A}^i_{\text{mar}}$ erweitert. Die Aktion no-op definiert dabei die Aktion unter der kein Handelsangebot gemacht werden soll. Zu jedem Zeitschritt werden die tatsächlichen Transaktionen auf Basis der Kaufangebote der Agenten durch die Marktfunktion berechnet, die in der Matrix $S \in N \times N$ gespeichert werden. Im Falle des Action Trading sind keine expliziten Verkaufsangebote zu speichern, da sich die Verkaufsangebote aus den ausgeführten Aktionen der Agenten in dem jeweiligen Zeitschritt ergeben.

4.2.5.1.1 Wachstum Aktionsraum Durch die Erweiterung der Aktionsräume in der oben beschriebenen Art und Weise, wachsen die erweiterten Aktionsräume $\mathcal{A}^i_{\text{SMS}}$ des stochastischen Spiels stark mit der Anzahl der verfügbaren atomaren Aktionen $|\mathcal{A}^i|$ sowie mit der Anzahl der Agenten $|\mathcal{N}|$. Ohne weitere Einschränkungen beträgt die Mächtigkeit des erweiterten Aktionsraums $\mathcal{A}^i_{\text{SMS}} = |\mathcal{A}^i| \times (|\mathcal{A}^i| + 1)^{|\mathcal{N}|}$. Für das Coin Game bedeutet das, dass der erweiterte Aktionsraum im Falle zweier Agenten bereits eine Mächtigkeit von $|\mathcal{A}^i_{\text{SMS}}| = 4 \times 5 = 20$ hat. Würde ein weiterer Agent hinzukommen würde sich die Mächtigkeit des Aktionsraums dadurch schon auf $|\mathcal{A}^i_{\text{SMS}}| = 4 \times 5 \times 5 = 100$ belaufen. Durch das exponentielle Wachstum des Aktionsraums können sich somit schon für Multiagenten-Umgebungen mit wenigen Agenten Schwierigkeiten für das Training mittels *deep* Reinforcement Learning ergeben [22].

In der Praxis kann es sich daher anbieten das Wachstum der Aktionsräume durch einschränkende Annahmen zu limitieren. Eine Möglichkeit das Wachstum im Falle mehrerer Agenten effektiv zu begrenzen, besteht darin pro Agent nur ein Handelsangebot pro Zeitschritt für einen anderen Agenten zu erlauben. Dadurch reduziert sich das Wachstum auf $|\mathcal{A}^i_{\text{SMS}}| = |\mathcal{A}^i| + (|\mathcal{A}^i| * |\mathcal{A}^i| \times (|\mathcal{N}| - 1))$. Weitere Möglichkeiten den Aktionsraum zu limitieren, bestehen darin, nur Untermengen der möglichen atomaren Aktionen als Handelsaktionen zu erlauben oder atomare Aktionen zu bündeln, so dass nicht einzelne Aktionen gehandelt werden, sondern ein Bündel gekauft werden kann, aus dem dann eine Aktion (zufällig) zur Ausführung gewählt werden kann.

4.2.5.1.2 Leistungszeitpunkt Mit Action Trading wählt Agent i im erweiterten stochastischen Spiel eine Aktion $a^i_t = (a^{i,\text{env}}_t, a^{i,\text{mar},j}_t)$, wobei $a^{i,\text{env}}_t \in \mathcal{A}^i$ die eigene Aktion zur Ausführung ist und $a^{i,\text{mar},j}_t \in \mathcal{A}^i_{\text{mar}}$ die Aktion beschreibt, die i für Agent j zum Handel zum Zeitpunkt t vorschlägt. Theoretisch können somit auch Aktionen zum Handel vorgeschlagen werden, die bereits n Zeitschritte in der Vergangenheit liegen $(t-n)$ oder für einen beliebigen Zeitpunkt in der Zukunft definiert werden $(t+n)$. In dieser Arbeit wurde die Möglichkeit des zeitlich versetzten Erbringens der Leistung eines Handels nicht weiter untersucht. Das Kaufangebot und das Erbringen der dadurch definierten Leistung des anderen Agenten müssen somit in dem gleichen Zeitschritt erfolgen. Durch diese Modellierung finden die Transaktionen zwischen den Agenten nicht sequentiell statt, sondern die Leistungserbringung des einen Agenten fällt zeitlich mit dem Kaufangebot zusammen.

4.2.5.1.3 Preis Wie in Abschnitt 4.2 erfordert die Erweiterung des stochastischen Spiels in ein stochastisches Spiel mit Markt, neben der Erweiterung der Aktionsräume auch das Setzen eines Preises p. Für den Rahmen dieses Kapitels wird der Preis als Teil der frei zu setzenden Parameter betrachtet und ist nicht Teil des Lernproblems, das die Agenten selbst erlernen. Zur Modellierung eines Verhandlungsprozesses zum Finden eines Preises siehe Abschnitt 3.3. Für den Fall des Coin Games wird in allen Experimenten $p = 1,25$ gesetzt, da die-

ser Wert den möglichen Gewinn eines Agenten durch das Einsammeln einer Münze übersteigt (+1) und geringer ist als der Verlust, den der designierte Besitzer der Münze erleiden würde (−2), wenn der andere Agent die Münze einsammelt. Somit liegt p im Bereich der Verhandlungsmenge möglicher Preise und stellt für beide Agenten eine Möglichkeit zur individuellen Verbesserung dar. Beide Agenten sollten $p = 1,25$ daher für eine Transaktion akzeptieren.

4.2.5.1.4 Anzahl möglicher Transaktionen Die Integration der Handelsaktionen erlaubt bilateralen Handel von Aktionen zwischen den Agenten. In der Theorie könnte jeder Agent mit jedem anderen Agenten pro Zeitschritt genau zwei Transaktionen durchführen, somit ist die Anzahl der Transaktionen pro Zeitschritt auf $2 \times |\mathcal{N}|$ beschränkt. Die Anzahl der möglichen Transaktionen lässt sich durch den Parameter zur Steuerung des Handelsbudgets m ggf. weiter limitieren. Zur Evaluation des Coin Games wird ein Handelsbudget pro neu erschienener Münze gesetzt, so dass pro Münze zwei Transaktionen möglich sind. Bei einem Preis $p = 1,25$ entspricht das einem Budget von $m = 2,5$.

4.2.5.2 Experimente

In diesem Abschnitt werden nun die durchgeführten Experimente im Coin Game in den Varianten mit zwei Spielern und mit vier Spielern beschrieben. Die Variante mit vier Spielern hat dabei das Ziel die Erweiterbarkeit des Ansatzes auf mehr als zwei Agenten zu untersuchen. Das Spiel wird dazu so erweitert, dass jeder der vier Agenten eine ihm zugewiesene Münze hat. Sammelt ein anderer Agent diese Münze, so erhält der eigentliche Besitzer der Münze eine Bestrafung. Die absolute Höhe der Belohnungen und Bestrafungen bleibt unverändert. Es werden jeweils zwei Konfigurationen des Coin Games verglichen: zuerst werden die Experimente ohne Markt durchgeführt, anschließend wird der Marktmechanismus in Form des Action Trading integriert und das gleiche Experiment erneut durchgeführt, wobei alle marktunabhängigen Parameter gleich belassen werden.

4.2.5.2.1 Agenten und Training Die Agenten im Coin Game werden jeweils durch eine unabhängige Instanz des Algorithmus *Deep Q-Network* (DQN) [57] repräsentiert. Dabei wird im Vergleich zu der originalen DQN Implementierung eine andere Netzarchitektur gewählt. Der Grund dafür liegt in dem verhältnismäßig kleinen Beobachtungsraum, den die Agenten im Coin Game, im Vergleich zu den auf Pixel-Daten basierten Atari-Spielen, haben (für Atari wurde DQN auf reinen Pixel-Daten im Format $84 \times 84 \times 3$ trainiert, so dass der Einsatz eines *Convolutional Neural Nets* das Training vereinfachte). Das hier eingesetzte Trainingsverfahren basiert nicht auf Pixel-Daten, sondern auf einer niedrigdimensionalen Repräsentation des Zustands s_t. Die Beobachtung $O(s_t)$ eines Agenten umfasst dabei die eigene Position im Grid p_i, die Position aller anderen Agenten p_j für $j \in \mathcal{N} \setminus i$, sowie den aktuellen Typ y_i der Münze

mit $i \in \mathcal{N}$ sowie die Position der aktuellen Münze im Grid.

Zur Repräsentation der Q-Funktion wird ein *Feedforward*-Netz mit zwei versteckten Schichten und 64 Gewichten pro Schicht verwendet. Zum Training des Netzes speichert jeder Agent Transitionen der Form (s, a, r, s') in seinen Experience Replay Speicher. Die Größe des Speichers ist auf 50.000 Transitionen begrenzt, so dass ältere Transitionen im Zeitablauf aus dem Speicher entfernt werden. Zum Training in einem Zeitschritt wird eine *Minibatch* aus zufälligen Transitionen aus dem Speicher entnommen und damit ein stochastischer Gradienten-Schritt gemäß der Verlustfunktion:

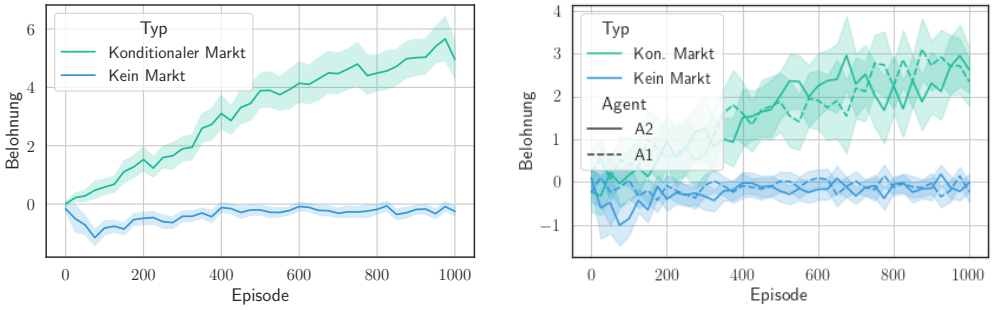
$$\mathcal{L}(\theta) = [r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)]^2$$

durchgeführt, wobei θ^- die Gewichte des *Target*-Netzes und θ die Gewichte des *Policy*-Netzes beschreibt. Zum Training der Neuronalen Netze wird eine Lernrate von $\alpha = 0,001$ verwendet. Zur Diskontierung wird in allen Experimenten der Wert $\gamma = 0,99$ verwendet. Durch die Integration des Marktes und die damit einhergehende Veränderung der Aktionsräume ist eine leichte Modifikation der Netzarchitektur notwendig. Dazu wird jedoch lediglich die letzte Schicht (Ausgabeschicht) des neuronalen Netzes entsprechend erweitert, so dass die Anzahl der Ausgänge der Anzahl der Aktionen entspricht. Die restliche Netzarchitektur bleibt unverändert. Die dargestellten Ergebnisse enthalten jeweils 80 Trainingsläufe, wobei in den Abbildungen stets die Mittelwerte sowie die Konfidenzintervalle zu einem Konfidenzniveau von 95% dargestellt sind.

Die Episoden im Coin Game haben eine feste Länge von 100 Zeitschritten. Jede Episode beginnt dabei mit einer zufälligen Konfiguration des Spielfelds, dazu gehören die Startpositionen der Agenten sowie die Position und der Typ der ersten Münze. Es erscheint jeweils nur eine Münze gleichzeitig. Sobald die aktuelle Münze gesammelt wurde, wird eine neue Münze an einer zufälligen Position des Spielfelds erzeugt. Treten beide Agenten zum gleichen Zeitpunkt auf das Feld einer Münze, so wird der Konflikt zufällig aufgelöst.

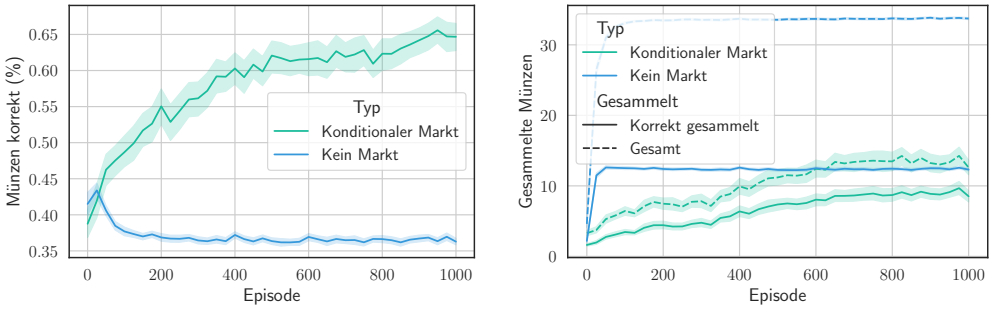
4.2.5.2.2 Ergebnisse zwei Agenten Abbildung 4.9 zeigt die Entwicklung der Belohnungen über den Trainingsverlauf von 1000 Episoden. Die Gesamtbelohnung (Summe der Belohnungen beider Agenten) ist in Abbildung 4.9a dargestellt. Die Entwicklung der Gesamtbelohnung ohne Markt (dargestellt in blau) ändert sich über den Verlauf des Trainings nur geringfügig und kann zu keinem Zeitpunkt konsistent in den positiven Bereich kommen. Durch die Erweiterung mittels Action Trading (dargestellt in grün) kann die erzielte Gesamtbelohnung im Laufe des Trainings deutlich erhöht werden. Neben der erhöhten Gesamtbelohnung zeigt Abbildung 4.9b, dass beide Agenten in ähnlichem Maße von dem Markt profitieren, da sich die Belohnungen beider Agenten in der Variante mit Action Trading im Vergleich zu dem Coin Game ohne Markt deutlich steigern.

Die höhere erzielte Gesamtbelohnung resultiert aus einem höheren Anteil



(a) Gesamtbelohnung (b) Belohnung pro Agent

Abbildung 4.9: Vergleich der Belohnungen im Coin Game mit und ohne Action Trading.



(a) Gesamtbelohnung (b) Anzahl gesammelter Münzen

Abbildung 4.10: Anteil der korrekt gesammelten Münzen im Coin Game und Gesamtzahl gesammelter Münzen.

korrekt gesammelter Münzen, d.h. Münzen, die jeweils von dem designierten Besitzer gesammelt wurden. Der Anteil korrekt gesammelter Münzen über den Trainingsverlauf ist in Abbildung 4.10 dargestellt. Hier zeigt sich, dass im Falle eines Marktes gegen Ende des Trainings knapp zwei von drei Münzen korrekt gesammelt werden, wohingegen ohne Markt nur beinahe eine von drei Münzen korrekt gesammelt wird. Diese Diskrepanz führt bereits dazu, dass die Gesamtbelohnung ohne Markt erheblich geringer ausfällt, da für jede nicht korrekt gesammelte Münze eine (netto) negative Gesamtbelohnung von -1 entsteht.

Interessanterweise zeigt sich, dass der Anteil der insgesamt gesammelten Münzen im Falle ohne Markt deutlich höher ist, als mit Markt, wie in Abbildung 4.10b dargestellt. So werden ohne Markt bereits nach wenigen Trainingsepisoden über 30 Münzen gesammelt und mit Markt werden pro Episode bei Beendigung des Trainings nur ca. 12 Münzen gesammelt. Da der Anteil korrekt gesammelter Münzen mit Markt aber deutlich höher ist, reicht die verhältnismäßig geringe Gesamtzahl an gesammelten Münzen jedoch aus, eine

höhere Gesamtbelohnung zu generieren als ohne Markt. Anders ausgedrückt, erzeugen die Agenten somit mit weniger Aufwand einen höheren Gewinn, erreichen also einen höheren Grad an Effizienz.

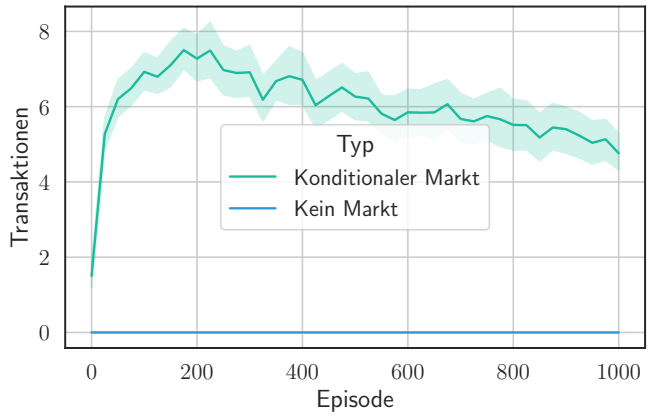
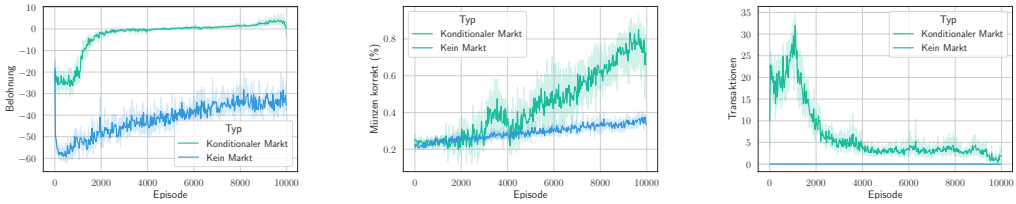


Abbildung 4.11: Entwicklung der Transaktionen pro Episode im Coin Game.

Das insgesamt bessere Ergebnis, das durch den Marktansatz generiert wird, spiegelt sich in den Transaktionen zwischen den Agenten wider. Die Anzahl der Transaktionen pro Episode, dargestellt in Abbildung 4.10b, steigt zu Beginn des Trainings steil an, ist jedoch im weiteren Verlauf des Trainings leicht rückgängig und liegt gegen Ende des Trainings bei ungefähr fünf Transaktionen pro Episode. Eine Transaktion beschreibt dabei einen tatsächlich durchgeführten Austausch von Aktion gegen Belohnung. Bei einem Preis von $p = 1,25$ entspricht das einem Handelsvolumen von 6.25 transferierten Belohnungseinheiten pro Episode. Die Tatsache, dass die Anzahl der Transaktionen zu Beginn stark ansteigt und sich anschließend rückläufig entwickelt, kann zum einen daran liegen, dass die Agenten erst im Laufe des Trainings lernen zu handeln und somit am Anfang — auch bedingt durch die stark zufällig geprägte Aktionswahl — zu viele (ineffiziente) Transaktionen durchführen. Ein weiterer möglicher Effekt ist die bereits erfolgte Konditionierung des Gegenübers, so dass eine tatsächliche Transaktion im Verlauf weniger notwendig wird um den anderen Agenten zu incentivieren (siehe Diskussion dazu in Abschnitt 4.2.5.3).

4.2.5.2.3 Ergebnisse mit vier Agenten Zur Untersuchung des Action Trading in einem sequentiellen sozialen Dilemma mit mehr als zwei Agenten, wird das Coin Game zu einer Variante mit vier Agenten erweitert. Die Regeln bleiben dabei analog zu der Variante mit zwei Spielern, d.h. ein Agent erhält eine negative Belohnung von -2, wenn ein anderer beliebiger Agent dessen Münze einsammelt, und ein Agent erhält eine Belohnung von $+1$ für das Einsammeln einer beliebigen Münze. Die Trainingsdauer wird im Vergleich zu zwei Spielern von 1000 Episoden auf 10.000 Episoden erhöht, da die erhöhte Komplexität durch mehr Spieler einen höheren Trainingsaufwand erfordert. Die Netzarchi-



(a) Gesamtbelohnung (b) Gesammelte Münzen (c) Transaktionen

Abbildung 4.12: Ergebnisse im Coin Game mit vier Agenten.

tekturen sowie die Belegung der Hyper-Parameter verändern sich hingegen nicht und werden, wie im Fall von zwei Spielern, verwendet.

Abbildung 4.12 zeigt die Ergebnisse im Coin Game mit vier Agenten. Die Gesamtbelohnung ist zu Beginn des Trainings im negativen Bereich, sowohl für die Variante mit Markt als auch ohne Markt. Im Verlauf des Trainings steigt die Belohnung für den Fall mit Markt jedoch in einen positiven Bereich, wohingegen die Belohnung ohne Markt zunächst weiter stark abfällt und sich anschließend wieder in Richtung des Ausgangsniveaus entwickelt. Der Anteil korrekt gesammelter Münzen entwickelt sich in beiden Varianten positiv, jedoch kann im Falle des Marktes ein deutlich höherer Anteil erreicht werden von beinahe 80%, wohingegen der Anteil korrekt gesammelter Münzen ohne Markt unter 40% bleibt. Die Entwicklung der Transaktionen zeigt ein ähnliches Muster wie im Fall von zwei Agenten: nachdem die Anzahl der getätigten Transaktionen zunächst stark ansteigt, fällt sie im Laufe des Trainings wieder auf ein deutlich niedrigeres Niveau ab, so dass gegen Ende des Trainings unter 5 Transaktionen pro Episode getätigt werden.

4.2.5.3 Diskussion

Abschließend werden in diesem Abschnitt die Ergebnisse der Experimente diskutiert. In den vorgestellten Ergebnissen konnte gezeigt werden, dass die Integration eines konditionalen Marktes in Form von Action Trading sich positiv auf die Gesamtbelohnung auswirkt und auch den einzelnen Agenten einen Vorteil gegenüber der Situation ohne Markt verschaffen kann.

4.2.5.3.1 Preise Der Handel zwischen zwei Agenten in den dargestellten Experimenten wird möglich durch die Wahl eines Preises, der die Bedingung erfüllt, dass beide Agenten sich durch die Transaktion besser stellen können. Für das Coin Game wurde dieser Preis mit $p = 1.25$ gesetzt. Die Wahl des Preises in diesem Fall resultiert aus der Tatsache, dass durch eine Transaktion, die das Einsammeln der Münze von Agent i durch Agent j verhindert, Agent i ein Schaden in Höhe von -2 erspart bleibt und Agent j eine Belohnung von $+1$ entgeht. Somit wäre Agent i bereit bis zu 2 Belohnungseinheiten zu zahlen, um Agent j vom Einsammeln der Münze abzuhalten, wohingegen Agent j

mindestens 1 Belohnungseinheit erhalten müsste, um sich vom Einsammeln abbringen zu lassen. Die Wahl von $p = 1.25$ ist somit willkürlich im Intervall $[1, 2]$ gewählt worden und ein anderer Preis in diesem Intervall sollte theoretisch ebenso funktionieren. Eine Frage, die daraus resultiert ist, inwiefern die Wahl des Preises auch durch die Agenten selbst vorgenommen werden kann, d.h. inwieweit sich das Finden eines Gleichgewicht-Preises als Teil des Lernproblems modellieren lässt, so dass kein manuelles Setzen des Preises mehr notwendig ist. Die Modellierung eines solchen Verhandlungsprozesses mittels Reinforcement Learning wird in Abschnitt 3.3 behandelt.

4.2.5.3.2 Handels-Budget Ein weiterer Parameter, durch den sich die Handelsaktivität der Agenten regulieren lässt, ist das Budget m_i, das die mögliche Anzahl an Transaktionen pro Episode für Agent i definiert. Im Falle des Coin Games wurde die Anzahl der möglichen Transaktionen pro erschienener Münze durch ein Budget $m_i = 2.5$ begrenzt, wodurch bei einem Preis von $p = 1.25$ genau zwei Transaktionen pro Agent und pro neuer Münze möglich sind. Die Wahl eines limitierenden Handels-Budgets ist motiviert durch die Beobachtung, das in früheren Experimenten ohne Budget eine Art pathologisches Handelsverhalten zwischen den Agenten entstand, bei dem die Agenten das eigentliche Ziel nicht mehr verfolgten, sondern lediglich über Transaktionen Belohnungen untereinander austauschten. Die Vermutung ist, dass das eigentliche Lernziel durch die gehandelten Belohnungen überschattet wird. Durch die Einführung eines limitierenden Budgets konnte dieser Effekt vermieden werden und die Agenten lernten nun sowohl miteinander zu handeln als auch das eigentliche Lernziel (das Sammeln der Münzen) zu verfolgen.

4.2.5.3.3 Konditionierungs-Effekte Wie in den Abbildungen 4.11 und 4.12c kommt es sowohl im Falle von zwei Agenten als auch bei vier Agenten dazu, dass die Anzahl der Transaktionen zu Beginn des Trainings stark ansteigt und anschließend im Verlauf des Trainings wieder abnimmt. Der starke Anstieg zu Beginn lässt sich dadurch erklären, dass die Agenten anfänglich sehr stark explorieren, d.h. Aktionen zufällig wählen. Da dadurch Agent i auch sehr häufig Handelsangebote an Agent j richtet, lernt Agent j sehr schnell, dass er für bestimmte Aktionen von dem anderen Agenten belohnt wird. Agent i wiederum lernt, dass eine Transaktion zu einer negativen Belohnung von -1.25 führt, wodurch er die Handelsaktionen beginnt zu vermeiden. Gleichzeitig verschafft Agent i sich durch erfolgreiche Transaktionen womöglich eine höhere Chance auf das Sammeln der aktuellen Münze, wodurch das Handeln (indirekt) positiv bestärkt wird. Ein weiterer Aspekt dabei ist, dass Agent j über den Verlauf des Trainings zu bestimmten Verhaltensweisen konditioniert wird, die aber nicht mehr notwendigerweise von Agent i belohnt werden. So hat Agent j möglicherweise gelernt, dass es häufig zu einer Belohnung geführt hat, wenn er sich von der Münze des anderen Agenten entfernt hat. Somit wird er sich auch zukünftig mit erhöhter Wahrscheinlichkeit von der Münze fern-

halten. Agent i entdeckt durch sein exploratives Vorgehen aber nun, dass es nicht mehr notwendig ist eine Belohnung an j für diese Verhaltensweise zu transferieren, da j dieses Verhaltensmuster auch ohne diese Bestärkung zeigt. Agent j ist also bereits auf dieses Verhaltensmuster konditioniert. Setzt i nun zu lange mit den Zahlungen an j aus, wird j wiederum merken, dass sich nun das Einsammeln der (fremden) Münze wieder mehr lohnt als ihr fern zu bleiben, wodurch i wieder einen Anreiz bekommt die Zahlungen wieder aufzunehmen. Es entsteht ein Zyklus, dessen Durchlaufgeschwindigkeit von der Adaptionsgeschwindigkeit (d.h. auch der Lernrate) der Agenten abhängt. In den gezeigten Experimenten konnte ein Wechsel zwischen diesen Phasen durch ein kontinuierliches Abkühlen der Explorationsrate unterbunden werden.

4.3 Bedingungsloser Markt: Shareholding

Nachdem im vorherigen Abschnitt der konditionale Marktmechanismus Action Trading erläutert wurde, folgt in diesem Abschnitt die Beschreibung eines bedingungslosen (nicht-konditionalen) Ansatzes. Ähnlich wie im Fall des Action Trading wird eine konkrete bedingungslose Marktform, das sogenannte Shareholding beschrieben. Die Evaluation erfolgt in einem eigenen Abschnitt, wobei auch der Vergleich zum Action Trading gezogen wird.

4.3.1 Idee und Motivation

Wie in Abschnitt 3.2.3 beschrieben, beruht die Idee der bedingungslosen Marktmechanismen auf dem Prinzip des Wertpapierhandels sowie es von börsennotierten Unternehmen bekannt ist. Ein börsennotiertes Unternehmen hat die Möglichkeit Unternehmensanteile über spezielle Marktplätze (Börsen) zu verkaufen und somit einen Teil der Finanzierung, beispielsweise für Investitionen, zu akquirieren. Für den Käufer kann sich der Erwerb von Unternehmensanteilen in unterschiedlicher Form lohnen: Einerseits gibt es für die Unternehmensanteile meistens eine Dividende, also eine monetäre Auszahlung an alle Anteilseigner, deren Höhe sich an den aktuellen Unternehmensgewinnen orientiert. Andererseits kann sich der Besitz von Unternehmensanteilen auch dadurch lohnen, dass der Wert der Anteile im Zeitverlauf steigt. In diesem Fall können die Unternehmensanteile zu einem späteren Zeitpunkt wieder zu einem höheren Preis an der Börse veräußert werden. Es wirkt sich somit sowohl der erwirtschaftete Gewinn als auch einer bessere Gesamtbewertung des Unternehmens, positiv auf die finanzielle Situation der Anteilseigner aus.

Das Unternehmen und der Anteilseigner gehen somit eine Bindung ein, durch die deren Interessen angeglichen werden. Durch den Erwerb von Anteilen ist es im Interesse des Aktionärs, dass das Unternehmen dessen Anteile gehalten werden, seine betriebswirtschaftlichen Ziele erreicht. Würde man sowohl das Unternehmen als auch die Anteilseigner als Entitäten eines Multiagentensystems definieren, so könnte man sagen, dass durch den Erwerb von Unterneh-

mensanteilen die Belohnungsfunktion des Unternehmens positiv mit der Belohnungsfunktion des Anteilseigners in Korrelation gebracht wird. Im Extremfall, wenn alle Agenten ihre Belohnung nur noch aus der Belohnung eines einzigen Agenten (dem Unternehmen) generieren würden, so ließe sich ein zuvor gemischt-motiviertes Szenario in ein rein kooperatives Szenario transformieren. Die Idee des in diesem Abschnitt vorgestellten Konzepts ist es daher, dass die Agenten durch den Kauf und Verkauf von Anteilen ihrer eigenen Belohnungen, ihre Interessen angleichen können, so dass die Ziele derjenigen Agenten priorisiert werden, die die höchste *Rendite* versprechen.

4.3.2 Verwandte Arbeiten

Im Folgenden wird ein Überblick über verwandte Arbeiten zu dem Konzept des Shareholding gegeben, darunter die Verwendung von Metaheuristiken, Modifikation der Belohnungsfunktionen, Social Learning und Partnerwahl.

4.3.2.0.1 Metaheuristiken und Belohnungs-Modifikationen In Umgebungen mit knappen Ressourcen und dezentral organisierten Einheiten kann es, selbst in kooperativen Systemen, bedingt durch die dezentrale Entscheidungsfindung, zu ineffizienten Ressourcenallokationen kommen. Bazzan schlägt daher einen zweigeteilten Prozess zur Entscheidungsfindung vor [11]: es gibt sowohl einen globalen Optimierungsprozess, der auf Grundlage von Metaheuristiken versucht ein globales Optimum zu finden, als auch dezentral lernende Komponenten, die durch Reinforcement Learning individuell optimale Policies erlernen. Beide Seiten tauschen mögliche Lösungen untereinander aus, die dann jeweils in dem Lösungsraum mitberücksichtigt werden.

Inspiriert durch die Entstehung kooperativer Verhaltensweisen in biologischen Prozessen, die sich basierend auf evolutionären Mechanismen entwickeln, haben Wang et al. [90] einen Ansatz vorgestellt, bei dem natürliche Selektion und Reinforcement Learning kombiniert werden, um Kooperation auf Agenten-Ebene zu erlernen. Dabei entwickeln die Agenten eine intrinsische Motivation, wobei die intrinsische Motivation selbst eine Funktion ist repräsentiert durch ein neuronales Netz, dessen Parameter durch einen evolutionären Prozess erlernt werden. Die Idee, die Belohnungen der Agenten zu modifizieren, um Kooperation zu erzeugen, wird auch von Mguni et al. aufgegriffen [53]. Dabei wird der zusätzliche Belohnungsterm jedoch nicht durch einen evolutionären Prozess erlernt, sondern durch eine Instanz (genannt *Incentive Designer*) vorgegeben. Der Incentive Designer beruht auf einem Bayesian Optimierungsverfahren, das die optimalen Incentives, basierend auf Simulationen, approximiert.

In dieser Arbeit werden keine expliziten Metaheuristiken verwendet, um die Ziele der Agenten anzugleichen. Vielmehr ergeben sich kooperative Strategien, basierend auf lokalem Handel von Anteilen.

4.3.2.0.2 Social Learning und Partnerwahl Chauduri et al. sehen Reinforcement Learning in der gewöhnlichen Ein-Agenten-Variante als ungeeignet, um kooperative Policies in Multiagentensystemen zu lernen [19]. Sie schlagen einen Ansatz vor, bei dem Kooperation mittels Reinforcement Learning durch das Imitieren von Aktionen anderer Agenten erlernt wird (genannt *Social Learning*). Der dadurch erzielte Zuwachs an Lerngeschwindigkeit, hilft auch dabei Kooperation in Szenarien mit geteilten Ressourcen zu erlernen. Ein weiterer Aspekt, der Einfluss auf die Entstehung kooperativer Verhaltensweisen in selbst-interessierten Systemen haben kann, ist die Möglichkeit der Agenten ihre Interaktionspartner frei zu wählen [4]. Dass die freie Partnerwahl auch im Rahmen des Multiagent Reinforcement Learning eine entscheidende Rolle spielt, zeigen Anastassacos et al. durch ein Experiment, bei dem die Agenten, basierend auf der Information wie die Interaktion in der Vorrunde mit einem gewissen Partner verlief, eine Partnerwahl treffen können [4]. Im Gegensatz zu zufällig gewählten Partnern, zeigt sich sowohl ein verbessertes soziales Gesamtergebnis als auch die Entstehung von kooperativeren Strategien zwischen den Agenten. Aspekte des Social Learning werden in dieser Arbeit nicht weiter aufgegriffen, da die Interaktionen nicht Teil der Modellierung dieser Arbeit sind. Vielmehr ergeben sich die Interaktionen aus den Bedingungen und Anreizsystemen, mit denen die Agenten konfrontiert werden.

4.3.3 Konzept

Im Folgenden wird nun der bedingungslose Marktmechanismus *Shareholding* beschrieben, der auf dem Prinzip von frei handelbaren Belohnungsanteilen beruht, wie in [76] vorgestellt. Bei diesem Ansatz kann jeder Agent $i \in \mathcal{N}$ als Emittent, d.h. als Veräußerer von Anteilen, in Erscheinung treten. Ausgegeben werden können Anteile an der zukünftigen (erwarteten) Belohnung des Agenten. Erwirbt ein Agent i also einen Belohnungsanteil eines anderen Agenten j, so wird i mit einem festen Anteil an den zukünftigen Belohnungen von j beteiligt.

Abbildung 4.13 veranschaulicht das Prinzip des Erwerbs und Verkaufs von Anteilen. Jeder Agent kann dabei über seine Anteile frei verfügen, kann also zu jedem Zeitpunkt entscheiden, ob Anteile veräußert werden sollen. Zusätzlich kann jeder Agent entscheiden, ob und von welchen anderen Agenten zu einem bestimmten Zeitpunkt, Anteile erworben werden sollen. Ein wesentlicher Unterschied im Vergleich zum realen Handeln von Unternehmensanteilen besteht darin, dass es nicht das Ziel des Handelns ist dem Emittenten durch den Verkauf von Aktien die Finanzierung anfallender Investitionen zu ermöglichen oder sich direkt durch den Verkauf zu bereichern. Vielmehr dient der Mechanismus dazu die Belohnungsfunktionen der Agenten miteinander zu korrelieren, so dass die Ziele der Agenten miteinander in Einklang gebracht werden. Der erzielte Preis, der für einen Anteil an der Belohnungsfunktion eines Agenten gezahlt wird, ist daher nicht die primäre Motivation eines Agenten, Anteile aus-

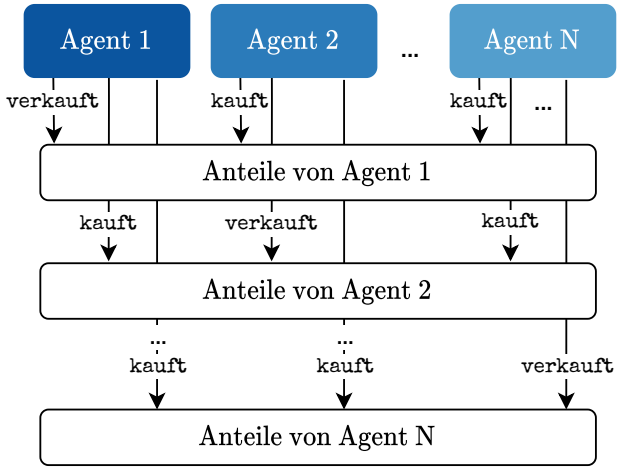


Abbildung 4.13: Durch das *Shareholding* wird es den Agenten ermöglicht Anteile der eigenen zukünftigen Belohnung an andere auszugeben und Anteile anderer Agenten zu erwerben.

zugeben. Im Idealfall sollte ein Agent sogar bereit sein, seine Anteile auch zu verschenken, wenn dadurch eine höhere eigene zukünftige Belohnung realisiert werden kann. Diese höhere Belohnung resultiert aus der erhöhten Kooperationsbereitschaft der anderen Agenten, die nun Anteilseigner sind und somit ebenfalls von der höheren Belohnung profitieren würden.

Algorithm 4 Marktfunktion: Shareholding

procedure SHAREHOLDERMARKET($\mathbf{a}_t, \mathbf{r}_t, s_t$)
  $B \leftarrow s_t$ ▷ Hole aktuelle Bilanz
  $K, V \leftarrow 0^{N \times N}$ ▷ Initialisiere Kauf- u. Verkaufsmatrix
  for agent $i \in N$ **do**
    $a_t^{i,\text{sell}}, a_t^{i,\text{buy},j} \leftarrow \mathbf{a}_t^i$ ▷ Handelsangebote von Agent i
    $V_i \leftarrow a_t^{i,\text{sell}}, K_{i,j} \leftarrow a_t^{i,\text{buy},j}$ ▷ Speichern in Handelsmatrizen
  $B \leftarrow V \wedge K$ ▷ Berechnung d. Transaktionen
  $\mathbf{r_t}' \leftarrow B, r_t, B$ ▷ Berechnung d. Belohnungen
  return $\mathbf{r_t}'$

Die Prozedur zur Berechnung der Transaktionen für den Handel mit Belohnungsanteilen ist in Algorithmus 4 dargestellt. Wie im Action Trading gibt es eine Bilanz $B \in \mathbb{N}^{|\mathcal{N}| \times |\mathcal{N}|}$, die die episodischen Forderungen und Verpflichtungen der Agenten gegeneinander abbildet. Der Eintrag $x_{i,j}$ der Matrix B kennzeichnet somit, dass Agent i eine Forderung gegenüber Agent j in Höhe von $x_{i,j} * p$ hat, wobei p dem festgelegten Wert eines Belohnungsanteils entspricht. Zu jedem Zeitschritt t werden die neuen Transaktionen aus dem Vektor aller Aktionen $\mathbf{a}_t$ ermittelt. Das Handeln von Belohnungsanteilen unterscheidet sich von dem in Abschnitt 4.2 vorgestellten Action Trading dadurch, dass die Transaktionen in diesem Ansatz an keine weiteren Bedingungen mehr gebun-

den sind, d.h. eine Transaktion ist bedingungslos. Jeder Agent i kommuniziert lediglich seine Bereitschaft Belohnungsanteile an Agent j zu verkaufen (durch die Aktion $a_t^{i,\text{sell},j}$) und seine Intention Anteile eines anderen Agenten j zu kaufen (durch die Aktion $a_t^{i,\text{buy},j}$). Beim Shareholding ist, ähnlich wie beim Action Trading, zu definieren mit welcher Granularität gehandelt werden kann, also ob Agenten definieren müssen an wen verkauft bzw. gekauft werden soll oder ob sich die Aktion *kaufen* bzw. *verkaufen* an alle potentiellen Handelspartner richtet (siehe dazu auch nächster Abschnitt).

Da eine Transaktion an keine weiteren Bedingungen gebunden ist, müssen nun auch nicht mehr die Aktionen der Agenten gespeichert und überprüft werden, um eine Transaktion zu validieren. Es wird für jeden Agenten $i \in \mathcal{N}$ überprüft, ob der Agent seine Anteile verkaufen möchte bzw. von welchem Agenten j Anteile gekauft werden sollen. Alle Kauf- und Verkaufsangebote zum Zeitpunkt t werden in den Matrizen $K, V \in \mathbb{N}^{|\mathcal{N}|\times|\mathcal{N}|}$ persistiert, wobei alle Verkaufsangebote in V und alle Kaufangebote in K gespeichert werden (theoretisch würde auch eine Matrix dafür genügen, doch der Eindeutigkeit wegen werden hier zwei Matrizen geführt). Wenn ein Verkaufs- und ein Kaufangebot zum Zeitpunkt t der Agenten i und j zusammenfallen, erwirbt der Käufer i einen Belohnungsanteil des Verkäufers j. Alle tatsächlich realisierten Transaktionen zum Zeitschritt t lassen sich dann durch Verundung von V und K ermitteln. Die aus dem Handel mit Belohnungsanteilen resultierenden neuen Belohnungen $\mathbf{r}'_t$ werden dann, basierend auf den neu hinzugekommenen Transaktionen B, den erzielten Belohnungen $\mathbf{r}_t$ zum Zeitschritt t und den über den Verlauf einer Episode bereits akkumulierten Belohnungsanteilen B, berechnet.

4.3.3.0.1 Aktionsraum Um den Agenten das Handeln mit Belohnungsanteilen zu ermöglichen, werden die Aktionsräume der Agenten erweitert. Dazu werden dem ursprünglichen (diskreten) Aktionsraum von Agent i $\mathcal{A}^o$ weitere Aktionen $\mathcal{A}^i_{\text{mar}} = \{\text{buy}, \text{sell}, \text{no-op}\}$ hinzugefügt. Um gleichzeitiges Handeln von Anteilen und ausführen von ursprünglichen Aktionen zu gewährleisten, wird das kartesische Produkt der Aktionsräume $\mathcal{A}^i$ und $\mathcal{A}^i_{\text{mar}}$ gebildet. Ähnlich wie bei dem Ansatz Action Trading, würde ohne weitere Einschränkungen der kombinierte Aktionsraum $\mathcal{A}_{SH}$, durch die Erweiterung mit den Handelsaktionen, sehr schnell mit der Anzahl der Agenten $|\mathcal{N}|$ anwachsen, da jeder Agent für jeden anderen Agenten entscheiden muss, ob mit diesem Agenten gehandelt werden soll. Um den Ansatz in Praxis daher skalierbar zu machen, wurde der Ansatz basierend auf folgender Beobachtung modifiziert: In den betrachteten Szenarien konkurrieren die Agenten um die gleichen Ressourcen, daher ist es für den Erfolg eines einzelnen Agenten essentiell, dass alle anderen Agenten bereit sind auf Ressourcen zu verzichten. Daher muss über den Verkauf eines Belohnungsanteils nicht für jeden Agenten individuell entschieden werden, sondern kann für alle Agenten simultan durchgeführt werden. Ein Agent, der sich für den Verkauf von Anteilen im nächsten Zeitschritt entschließt, ermöglicht daher allen anderen Agenten Anteile zu erwerben. Darüber hinaus können Agenten

jeweils nur ein einziges Verkaufs- oder Kaufangebot pro Zeitschritt machen, was die Größe des erweiterten Aktionsraums auf $|\mathcal{A}^i_{SH} = |\mathcal{A}^i| * |\{0,1\}| * |\mathcal{N}|$ reduziert.

4.3.3.0.2 Auszahlungen von Belohnungsanteilen Agenten können über den Verlauf einer Episode Anteile anderer Agenten ansammeln. Der Zeitpunkt der Auszahlung kann dann entweder zu einem bestimmten festen Zeitpunkt (Ende der Episode) erfolgen oder anhand der tatsächlich realisierten Belohnungen des zahlungspflichtigen Agenten ermittelt werden. Dabei ist festzulegen, ob Agenten nur selbst erworbene Belohnungen ausgeben können oder ob sie Belohnungen *erschaffen* können, d.h. Schulden gegenüber anderen Agenten können auch dann bezahlt werden, wenn selbst keine Belohnung in diesem Schritt generiert wurde. Da das Erschaffen von Belohnungen mit potentiellen Seiteneffekten verbunden ist (siehe dazu Abschnitt 3.2.7), wird hier nur der Fall betrachtet, bei dem ein Agent nur dann zahlungsfähig ist, wenn er eine positive Belohnung erhalten hat. Ist ein Agent zum Zeitschritt t nicht zahlungsfähig, so entsteht eine Verbindlichkeit gegenüber einem anderen Agenten, die beglichen wird, sobald der Agent zahlungsfähig wird. Es ist zu beachten, dass es Situationen geben kann, in denen Zahlungsverpflichtungen im Verlauf einer Episode nicht beglichen werden können. In jedem Fall lassen sich die Zahlungsverpflichtungen eines Agenten gegenüber seinen Anteilseignern, aus der Menge der vergebenen Anteile, bis zu einem bestimmten Zeitpunkt ermitteln.

4.3.3.0.3 Gültigkeit von Belohnungsanteilen Für das Verfahren ist auch zu entscheiden welche Laufzeit die erworbenen Belohnungsanteile der Agenten haben. So kann ein Belohnungsanteil nach der ersten erfolgten Auszahlung entweder aufgelöst werden, so dass keine Verbindung mehr zwischen den Agenten besteht. Alternativ kann ein Anteil eine längere Laufzeit haben, so dass Agenten kontinuierlich an den Belohnungen anderer Agenten teilhaben können. Für die Experimente, die im Rahmen dieser Arbeit durchgeführt wurden, wurden nur Belohnungsanteile betrachtet, die sich nach einmaliger Auszahlung wieder auflösen. In Abhängigkeit des untersuchten Szenarios, könnten aber andere Varianten ebenfalls sinnvoll sein. Auch eine Beschränkung der Laufzeit auf einzelne Episoden ist nicht zwangsläufig erforderlich. So wäre es vorstellbar, dass Agenten Belohnungsanteile episodenübergreifend halten können und somit langfristig partizipieren könnten. Auch der Weiterverkauf eines erworbenen Anteils wäre eine theoretische Möglichkeit, die in dieser Arbeit nicht weiter untersucht wurde.

4.4 Vergleich Action Trading und Shareholding

Im Folgenden Abschnitt werden nun die beiden Ansätze Action Trading und Shareholding miteinander verglichen. Dazu wird eine Evaluationsumgebung

eingeführt, die in der Anzahl der Agenten beliebig skaliert werden kann. Darüber hinaus erlaubt die Umgebung zwei unterschiedliche Modi, die sich bezüglich des Wettbewerbgrades, der zwischen den Agenten besteht, unterscheidet. In den darin durchgeführten Experimenten werden bis zu 16 Agenten, mittels Reinforcement Learning unabhängig und parallel trainiert.

4.4.1 Evaluationsumgebung

Zur Evaluation der beiden Marktansätze Action Trading und Shareholding wird ein Szenario verwendet, das einen industriellen Anwendungsfall für ein verteilt lernendes Multiagentensystem darstellt. Es wird ein Ressourcenallokations-Problem betrachtet, bei dem die Agenten um knappe Ressourcen konkurrieren. Inspiriert ist das Szenario von einer Produktionsanlage, in der autonome Einheiten individuelle Aufträge bearbeiten. Die autonomen Einheiten können dabei selbstlernende Roboter darstellen, die unterschiedlichen Firmen zugehören, die sich die gemeinsame Produktionsanlage aus Kostengründen teilen. Dargestellt wird die Fabrik durch ein zweidimensionales Grid, dessen Größe nach gewünschter Komplexität gesetzt werden kann. Im Folgenden werden zwei Varianten einer solchen Produktionsanlage vorgestellt, die sich in der Art und Weise, wie die Agenten miteinander in Beziehung stehen, unterscheiden. Im ersten Szenario wird angenommen die Agenten sind homogen, d.h., verfügen über die gleichen Fähigkeiten und Eigenschaften. Im zweiten Szenario gibt es zwei Klassen von Agenten, die unterschiedliche Fähigkeiten besitzen. Beide Varianten stellen die Agenten vor unterschiedliche Herausforderungen bezüglich potentieller Kooperation.

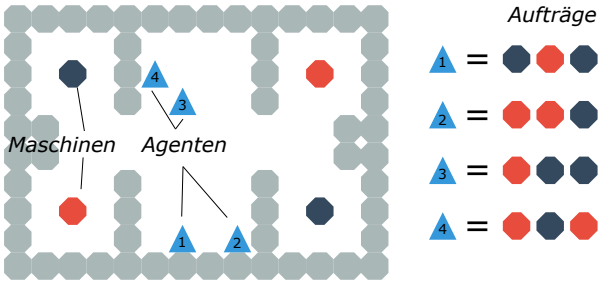
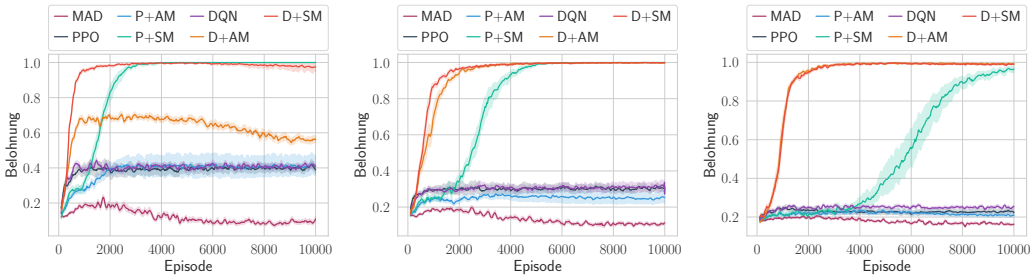


Abbildung 4.14: Die Smartfactory: die Agenten (blaue Dreiecke) müssen verschiedene Maschinen abfahren (rote bzw. schwarze Hexagone), um ihre individuellen Aufträge abzuarbeiten.

4.4.1.0.1 Smartfactory Die erste Variante der Evaluationsumgebung repräsentiert eine Smartfactory (*SF*), d.h. eine Produktionsanlage in der autonome Einheiten lernen individuelle Aufträge abzuarbeiten. Dazu enthält die Fabrik m verschiedene Maschinentypen, die von den Agenten genutzt werden können

(eine Visualisierung der Fabrik ist in Abbildung 4.14 zu sehen). In jeder neuen Episode erhält jeder Agent eine Aufgabe, die sich aus n Maschinen zusammensetzt, d.h. ein Agent muss alle n Maschinen aktivieren (indem er sie besucht), bevor der Auftrag als erledigt gilt. Ein Agent erhält erst dann eine Belohnung, wenn er alle Maschinen besucht hat, die zu seiner Aufgabe gehören. Es gibt zwei Parameter zur Feinabstimmung des Konfliktpotenzials zwischen den Agenten: Der Parameter t_{inactive} regelt die Anzahl der Zeitschritte, die eine Maschine nach der Nutzung inaktiv ist (und somit von keinem anderen Agenten genutzt werden kann). Folglich steigt die Konkurrenz zwischen den Agenten mit zunehmendem t_{inactive}. Der zweite Parameter betrifft die Dringlichkeit oder Priorität eines Agenten, seine Aufgabe zu erfüllen, mit zwei möglichen Prioritäten (hoch oder niedrig). Wenn die Aufgabe eines Agenten eine hohe Priorität hat, erhält dieser Agent nach der Erledigung eine höhere Belohnung r_{high} als ein Agent mit einer Aufgabe niedriger Priorität (der eine durch r_{low} definierte Belohnung erhält). Ein zunehmender Abstand zwischen r_{high} und r_{low} erhöht somit den potenziellen Nutzen der Koordination und Kooperation zwischen den Agenten in Bezug auf die Gesamtbelohnung, da durch die begrenzte Episodenlänge nicht alle Aufträge abgearbeitet werden können. In allen Experimenten wird die Belohnung für die Abarbeitung eines Jobs mit hoher Priorität auf $r_{\text{high}} = 5$ gesetzt und für die Abarbeitung eines Jobs mit niedriger Priorität auf $r_{\text{low}} = 1$. Nach der Nutzung einer Maschine ist die Maschine für $t_{\text{inactive}} = 8$ Zeitschritte nicht nutzbar.

4.4.1.0.2 Raffinerie Die zweite Variante der Produktionsanlage (genannt Raffinerie (RF)) ist durch einen industriellen Raffinerie-Prozess inspiriert, d.h. in dieser Anlage können bestimmte Rohstoffe veredelt werden, wodurch eine Weiterverarbeitung möglich wird. In diesem Fall gibt es zwei Arten von Agenten, nämlich Agenten, die den verfügbaren Rohstoff raffinieren können und Agenten, die den raffinierten (veredelten) Rohstoff konsumieren können. Der Rohstoff tritt somit in zwei verschiedenen Klassen auf: unraffiniert und raffiniert. Das Unterscheidungsmerkmal zwischen den beiden Agententypen ist, dass die eine Gruppe von Agenten (Raffinierer) die Wahl hat, entweder die unraffinierte Ressource zu konsumieren, was zu einer geringen Belohnung r_{low} für den konsumierenden Agenten führt, oder die Ressource zu veredeln, wobei sie dann nicht mehr von ihm selbst konsumiert werden kann. Ein konsumierender Agent hingegen kann den Rohstoff nicht veredeln, sondern nur die bereits veredelten Einheiten konsumieren, was ihm eine verhältnismäßig hohe Belohnung r_{high} einbringt. Der Gesamtertrag wird also dadurch maximiert, dass alle verfügbaren Ressourcen zuerst veredelt und anschließend verbraucht werden. Das Dilemma in diesem Szenario besteht nun darin, dass jedoch ein raffinierender Agent das veredelte Gut nicht verbrauchen kann und es somit individuell rational ist, die Rohressource zu verbrauchen, um zumindest eine kleine Belohnung zu erzielen. Die Agenten müssten kooperieren, um den maximalen Ertrag aus den vorhandenen Ressourcen zu erzielen. In den durchgeführten Experimenten



(a) 4 Agenten (b) 8 Agenten (c) 16 Agenten

Abbildung 4.15: Erzielte Gesamtbelohnung in der Smartfactory für 4, 8 und 16 Agenten.

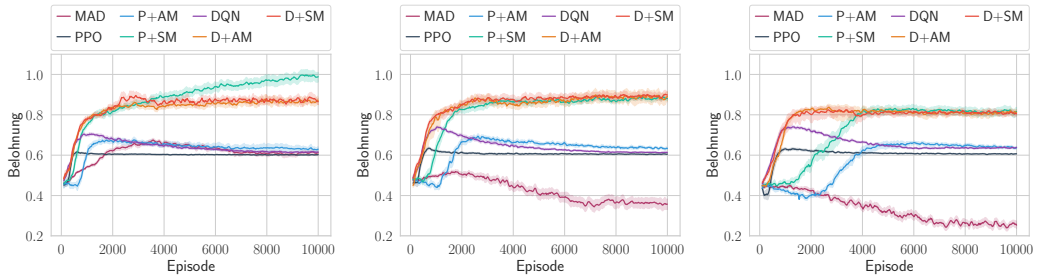
wird die Belohnung für den Konsum einer unveredelten Ressource $r_{\text{low}} = 0,02$ gesetzt und für den Konsum einer veredelten Ressource auf $r_{\text{high}} = 5$.

4.4.2 Agenten und Training

Um die beiden Marktmechanismen Action Trading und Shareholding zu vergleichen, werden sieben unterschiedliche Agententypen für die Evaluation herangezogen.

- *Deep-Q-Networks* (DQN) und *Proximal Policy Optimization* (PPO) ohne Marktansatz als unabhängige Referenzverfahren.
- *Deep-Q-Networks* und *Proximal Policy Optimization* (PPO) mit Action Trading wie in Abschnitt 4.2 beschrieben (abgekürzt D-AM bzw. P-AM).
- *Deep-Q-Networks* (DQN) und *Proximal Policy Optimization* (PPO) mit Shareholding wie in Abschnitt 4.3 beschrieben (abgekürzt D-SM bzw. P-SM).
- Zusätzlich werden Ergebnisse des *Multiagent Deep Deterministic Policy Gradient*-Algorithmus (MADDPG) [48] als weiteres Vergleichsverfahren gezeigt. MADDPG basiert auf einem zentralisierten Werte-Netz und ist daher kein rein dezentrales Verfahren (abgekürzt MAD).

Alle Experimente werden mit 4, 8 und 16 Agenten durchgeführt. Bei den DQN-Agenten wird das *Target*-Netz alle 500 Schritte aktualisiert und das Training beginnt nach anfänglichen 200 Schritten der Simulation. Für jeden Trainingsschritt wird eine *Mini-Batch* von 64 Erfahrungstupeln (s, a, r, s') aus dem *Replay-Memory* des Agenten gezogen. Während des Trainings wird die Exploration durch das ϵ-greedy Verfahren sichergestellt, wobei ϵ linear vom initialen Wert $1,0$ auf $\epsilon = 0,1$ herunter gekühlt wird. Zur Diskontierung wird $\gamma = 0,9$ verwendet. Die PPO-Agenten bestehen aus einem *Actor*- und einem *Critic*-Netz, mit zwei versteckten Schichten der Größe 32 und tangens hyperbolicus



(a) 4 Agenten (b) 8 Agenten (c) 16 Agenten

Abbildung 4.16: Erzielte Gesamtbelohnung in der Raffinerie-Umgebung für 4, 8 und 16 Agenten.

(tanh) als Aktivierungsfunktion. Die Modelle werden alle 5000 Schritte für 4 Epochen mit einer Lernrate von $0,002$ aktualisiert. Für diese Arbeit wurden die Hyperparameter für DQN und PPO nicht durch automatisierte Suchverfahren optimiert, sondern manuell angepasst, bis eine hinreichende Leistung in den Simulationen erreicht wurde. Anschließend wurden die auf diese Art ermittelten Parameter fixiert und für alle Verfahren identisch verwendet. Für das Verfahren MADDPG wurde zuerst ein Szenario, mit maximal zwei Agenten, zur Findung geeigneter Parameter verwendet. Diese Parameter wurden dann ebenfalls für die Trainingsläufe mit mehr als zwei Agenten verwendet.

4.4.3 Ergebnisse

Es folgen die Ergebnisse für die Smartfactory- und die Raffinerie-Umgebung, wobei die Ergebnisse für 4, 8 und 16 Agenten in Abbildung 4.15 (Smartfactory) und Abbildung 4.16 (Raffinerie) dargestellt sind. Zu sehen ist dabei die Gesamtbelohnung, d.h. die kumulierte Summe aller Belohnungen über alle Agenten. Jedes Verfahren wurde dabei über 30 unabhängige Trainingsläufe getestet, so dass die dargestellten Ergebnisse jeweils die Mittelwerte sowie Konfidenzintervalle zum Konfidenzniveau von 95% darstellen. Darüber hinaus wurden die Belohnungen auf den Bereich $0,1$ normalisiert, um alle Ergebnisse vergleichbar zu machen.

Sowohl in der Smartfactory als auch in der Raffinerie ist zu sehen, dass die Verfahren ohne Marktansatz, d.h. DQN, PPO und MADDPG, in allen Experimenten eine vergleichsweise geringe Gesamtbelohnung erzielen und sich im unteren oder mittleren Bereich der normalisierten Gesamtbelohnung bewegen. MADDPG schneidet dabei insgesamt am schlechtesten ab und es ist zu sehen, dass sich die erzielte Gesamtbelohnung über den Verlauf des Trainings zunehmend verringert, was dafür spricht, dass dieses Verfahren keine sinnvolle Policy für diese Umgebungen erlernen kann. Den insgesamt stabilsten Trainingsverlauf, bezogen auf die Konvergenzgeschwindigkeit und die erzielte Gesamtbelohnung, wird dabei aus der Kombination von DQN und Shareholding (D-SM) erzielt. Ausnahme bildet dabei in der Raffinerie-Umgebung die

Kombination aus PPO und Shareholding für vier Agenten, wo gegen Ende des Trainings das globale Optimum erreicht wird. Die Kombination von PPO und Shareholding (P-SM) erzielt zwar generell eine vergleichbare maximale Gesamtbelohnung wie DQN und Action Trading (D-AM), braucht dafür jedoch in allen Experimenten mehr Trainingszeit, um auf ein ähnliches Niveau zu kommen.

In der Smartfactory zeigt die Kombination aus DQN und Action Trading (D-AM) für die Experimente mit 4 Agenten eine geringere maximale Gesamtbelohnung und einen insgesamt etwas instabileren Trainingsverlauf als D-SM. Allerdings wird dieser Unterschied durch die Hinzunahme weiterer Agenten zunehmend geringer, so dass für 16 Agenten schlussendlich nahezu identische Trainingskurven entstehen. Im Unterschied dazu, kann durch die Kombination aus PPO mit dem Handeln von Belohnungsanteilen keine Verbesserung gegenüber reinem PPO erzielt werden.

Die Experimente in der Smartfactory- und Raffinerie-Umgebung zeigen, dass die Marktansätze generell zu einer höheren erzielten Gesamtbelohnung führen können. Allerdings gibt es Unterschiede im Ergebnis in Bezug auf die Kombination von Lernalgorithmus und Markttyp. Insgesamt konvergiert DQN dabei schneller und stabiler als PPO. Interessanterweise wird DQN mit Aktionsmarkt (D-AM) in der Smartfactory mit zunehmender Anzahl von Agenten stabiler. DQN-SM erreicht nahezu optimale Ergebnisse für alle Anzahlen von Agenten. PPO-SM erreicht ebenfalls nahezu optimale Ergebnisse, ist aber weniger effizient in Bezug auf die Trainingszeit, insbesondere bei einer höheren Anzahl von Agenten. Während DQN-AM mit zunehmender Anzahl von Agenten stabiler wird, nimmt die Leistung von PPO-AM in beiden Domänen ab, wenn mehr Agenten hinzukommen.

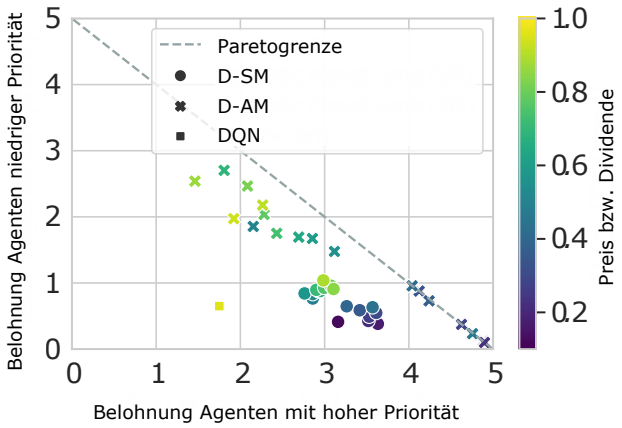


Abbildung 4.17: Darstellung der Aufteilung der erzielten Gesamtbelohnung auf Agenten mit niedriger Priorität und hoher Priorität, für die Smartfactory.

Neben der Betrachtung der Gesamtbelohnung, die durch die verschiedenen Ansätze erzielt wird, ist auch relevant, wie sich die Gesamtbelohnung auf die

Agenten des Systems verteilt. Für die Smartfactory lässt sich die Frage konkretisieren, indem untersucht wird, wie sich die Gesamtbelohnung auf Agenten mit niedrig priorisierten und hoch priorisierten Aufträgen verteilt. Dazu wurden jeweils für die Verfahren DQN, D-AM und D-SM eine Gruppierung der Belohnungen nach Auftragsart der Agenten (hoch priorisiert, niedrig priorisiert) vorgenommen. Da der Preis im Falle des Action Tradings bzw. die Dividende im Falle des Shareholdings ebenfalls einen Einfluss auf die erzielte Verteilung der Belohnungen haben kann, wurde die Analyse für verschiedene Preis- bzw. Dividenden-Niveaus für den Fall von 8 Agenten durchgeführt. Die Ergebnisse sind in Abbildung 4.17 dargestellt.

Jeder dargestellte Punkt (x, y) setzt sich aus dem Anteil x der Gesamtbelohnung von Agenten mit hoher Priorität und dem Anteil der Agenten mit niedriger Priorität y zusammen. Da die maximale Gesamtbelohnung bei 5 liegt, definiert die Linie von $(0, 5)$ nach $(5, 0)$ die Paretogrenze, d.h. für jeden Punkt auf dieser Linie gilt, dass es sich um ein optimales Ergebnis handelt. Je näher ein Punkt an der Paretogrenze liegt, desto besser ist das Ergebnis. Für alle getesteten Preise (Dividenden) liegen die Ergebnisse näher an der Pareto-Grenze (effizienter), wenn ein Markt vorhanden ist. Interessanterweise nimmt die Effizienz der Ergebnisse bei höheren Preisniveaus im Falle des Shareholdings ab, während die Ergebnisse weniger durch den Preis beim Action Trading beeinflusst werden. Schließlich beeinflusst das Dividenden- und Preisniveau die Aufteilung des Ertrags zwischen Agenten mit hoch- und niedrigprioren Aufgaben, so dass höhere Preise dazu führen, dass ein größerer Anteil des Ertrags den Agenten mit niedrigprioren Aufgaben zugewiesen wird.

4.4.4 Diskussion

Der Vergleich der beiden Verfahren Action Trading und Shareholding zeigt, dass der Erfolg eines Marktes stark von der Wahl des zugrundeliegenden Algorithmus abhängen kann. So bringt DQN in Kombination mit Action Trading in allen Experimenten bis auf die Variante mit vier Agenten in der Smartfactory die stabilsten Ergebnisse. DQN scheint also robust gegenüber der zunächst erhöhten Komplexität, bedingt durch den größeren Aktionsraum und dem damit einhergehenden Explorationsproblem. PPO scheint dagegen sensibler bezüglich der Wahl des konkreten Marktmechanismus zu reagieren. So bringt PPO mit Shareholding stabile Ergebnisse, braucht dafür aber im Vergleich zu DQN mehr Trainingsepisoden, insbesondere wenn es mehr Agenten werden. PPO kombiniert mit Action Trading kommt demgegenüber nicht über die Ergebnisse der Referenzverfahren ohne Marktmechanismus hinaus. Eine mögliche Erklärung dafür liegt in der unterschiedlichen Mächtigkeit der Aktionsräume der beiden Verfahren. So ist die Mächtigkeit des Aktionsraums unter Verwendung von Shareholding $|\mathcal{A}_{\text{SH}}| = |\mathcal{A}| * |\{0, 1\}| * |\mathcal{N}|$ so, dass für vier Agenten $|\mathcal{A}_{\text{SH}}| = 32$, für acht Agenten $|\mathcal{A}_{\text{SH}}| = 4 * 2 * 8 = 64$ und für 16 Agenten $|\mathcal{A}_{\text{SH}}| = 4 * 2 * 16 = 128$ gilt. Demgegenüber beträgt die Mächtigkeit des Akti-

onsraums durch das Action Trading $|\mathcal{A}^i_{\mathrm{AT}}| = |\mathcal{A}^i| + (|\mathcal{A}^i| * |\mathcal{A}^i| \times (|\mathcal{N}| - 1))$, was für vier Agenten zu $|\mathcal{A}^i_{\mathrm{AT}}| = 4 + (4 * 4 * 3) = 52$, für acht Agenten $|\mathcal{A}^i_{\mathrm{AT}}| = 116$ und für 16 Agenten zu $|\mathcal{A}^i_{\mathrm{AT}}| = 244$ Aktionen führt.

4.5 Zusammenfassung

In diesem Kapitel wurden zwei Formen von Marktmechanismen — konditionale und bedingungslose — anhand zweier konkreter Instanzen erläutert und evaluiert. Der konditionale Marktmechanismus, genannt Action Trading, erlaubt es Agenten die Transaktionen auf ausgeführte Aktionen zu konditionieren. Dadurch können die Agenten sich gegenseitig zu bestimmten Verhaltensweisen motivieren, indem sie Belohnungen gegen Aktionen tauschen. Der Ansatz wurde anhand eines Spiels mit zwei Spielern erläutert und anschließend auch in einem sequentiellen sozialem Dilemma, dem Coin Game, mit bis zu vier Agenten evaluiert.

Der bedingungslose Marktmechanismus ist von dem Prinzip des Aktienhandels inspiriert. Agenten könnten Anteile (Shares) ihrer eigenen zu erwartenden Belohnungen an andere Agenten ausgeben und dadurch auf deren zukünftige Unterstützung zur Erreichung ihrer eigenen Ziele hoffen. Der Agent, der einen Anteil erwirbt (Shareholder) erhält dadurch ebenfalls einen Teil der Belohnung. Somit lassen sich die Ziele der Agenten angleichen. Durch das Shareholding sollten somit die Ziele der Agenten mit der höchsten erwarteten Belohnung priorisiert werden, da alle durch deren Verwirklichung am meisten profitieren können.

Die beiden Ansätze wurden in einer parametrisierten Evaluationsumgebung evaluiert und miteinander verglichen. Die Umgebung stellt eine Produktionsanlage dar, die in zwei unterschiedlichen Varianten eingesetzt werden kann. Die erste Variante gleicht einer Smartfactory, in der Agenten unterschiedliche Maschinen zur Abarbeitung ihrer individuellen Aufträge benutzen können. Das zweite Szenario entspricht einer Raffinerie, wobei es zwei Typen von Agenten gibt, die um eine gemeinsame Ressource konkurrieren. In beiden Szenarien schneiden die Verfahren mit integriertem Markt am besten ab. Es gibt jedoch Unterschiede in dem erzielten Gesamtergebnis bezüglich des eingesetzten Lernalgorithmus und dem jeweiligen Marktmechanismus. Insgesamt wurden in den Experimenten bis zu 16 unabhängige Agenten trainiert. Bisher wurden noch keine Experimente mit mehr als 16 Agenten durchgeführt, jedoch lassen die Resultate noch keinen Effizienzrückgang erkennen, weshalb davon auszugehen ist, dass auch noch größere Experimente erfolgreich durchzuführen sind.

5 Vertragsbasierte Marktmechanismen und Kooperation in sozialen Dilemmas

In diesem Kapitel werden zwei weitere Austauschmechanismen vorgestellt, die die Ansätze aus Kapitel zwei komplementieren bzw. deren Anwendbarkeit auf eine weitere Klasse von Problemen untersuchen. Im ersten Abschnitt wird ein vertragsbasierter Marktmechanismus beschrieben. Hierbei können im Gegensatz zum konditionalen und bedingungslosen Handel, explizite Verträge für zukünftige Aktionen zwischen den Agenten vereinbart werden. Dadurch soll insbesondere den potentiellen negativen Effekten der Konditionierung, wie in Abschnitt 4.2.5.3 beschrieben, vorgebeugt werden. Durch das explizite Vereinbaren von Verhaltensvorschriften, ist einseitiges Abweichen nicht mehr möglich, wodurch für die Vertragspartner Sicherheit bezüglich des zu erwartenden Verhaltens des Gegenübers geschaffen wird.

Im zweiten Abschnitt dieses Kapitels wird die Anwendbarkeit der Mechanismen auf soziale Dilemmas mit symmetrischen Belohnungen untersucht. Hierbei zeigt sich, dass ein konditionaler Markt durch die Verwendung negativer Preise in einen Sanktionierungsmechanismus umgewandelt werden kann. Anstatt Agenten positiv durch Anreize zu bestimmten Verhaltensweisen zu motivieren, können die Agenten sich also bestrafen, d.h. Belohnungen voneinander abziehen. Es werden sowohl soziale Dilemmas mit zwei Spielern untersucht als auch eine N-Spieler-Variante des Prisoner's Dilemma. Der Sanktionierungsmechanismus wird dabei für bis zu 128 unabhängig trainierte Agenten evaluiert, wobei sich deutlich höhere Kooperationsgrade durch die Sanktionierungen erzielen lassen.

5.1 Vorveröffentlichungen

Die Kerninhalte dieses Kapitels wurden vom Autor bereits in [78] bzw. [75] publiziert. Die in diesem Kapitel dargestellten Abbildungen sind der Art nach ebenfalls bereits in [78] bzw.[75] veröffentlicht, wurden im Rahmen der Ausarbeitung dieser Arbeit aber neu visualisiert und beschriftet. Die Datengrundlage für die Abbildungen hat sich dabei jedoch nicht geändert und entspricht den gleichen Auswertungen wie in [78] bzw. [75] vorgestellt.

5.2 Vertragsbasierte Marktmechanismen

Im folgenden Abschnitt wird ein vertragsbasierter Marktmechanismus vorgestellt, der als *Distributed Emergent Agreement Learning* (DEAL) bezeichnet wird. Die dadurch realisierten Verträge zwischen zwei Agenten definieren sowohl eine Vereinbarung darüber, welche Aktionen für die Laufzeit des Vertrags ausgeführt werden sollen, als auch, welche Belohnungen den beteiligten Agenten durch das abschließen des Vertrags zuteil werden. Die Schätzung der Belohnungen basiert dabei auf dem Prinzip wertebasierter Kompensationen, wobei die Werte aus gelernten Q-Funktionen geholt werden. Der Ansatz wird in einer Variante der Smartfactory (siehe Abschnitt 4.4.1) mit bis zu 16 Agenten evaluiert.

5.2.1 Idee und Motivation

Die bisherigen marktbasierten Ansätze beruhen darauf, anderen Agenten Anreize zu schaffen, so dass sie sich zu Verhaltensweisen entscheiden, die für den anreizgebenden Agenten hilfreich sind. Das konditionale Handeln erlaubt es dabei, dass explizite Bedingungen definiert werden können, wie beispielsweise die Ausführung einer bestimmten Aktion, damit ein Transfer zwischen zwei Agenten zustande kommt. In diesem Abschnitt wird der konditionale Ansatz noch einen Schritt weitergeführt, indem Agenten nicht nur Bedingungen für einen Handel festlegen können, sondern explizite Verträge miteinander schließen können.

Dadurch lassen sich potentielle *Commitment*-Probleme lösen, die kooperative Verhaltensweisen verhindern können. Commitment-Probleme beschreiben die Unfähigkeit der Akteure glaubhafte Versprechen oder Drohungen auszusprechen. Das Prinzip lässt sich gut am Beispiel des Prisoner's Dilemma erläutern: Man stelle sich vor, beide Spieler könnten vor Spielbeginn miteinander kommunizieren. Dann könnten sie besprechen, dass sich beide für die kooperative Aktion entscheiden. Doch welche Sicherheit hat ein Spieler zum Zeitpunkt der Entscheidung, ob sich der andere Spieler auch daran hält? Die Spieler sehen sich also einem Commitment-Problem gegenüber, das aus dem Unvermögen resultiert glaubhafte Versprechungen (oder Drohungen) zu machen. Auch in Spielen ohne Nullsummen-Eigenschaft können Commitment-Probleme entstehen, wodurch kooperative Verhaltensweisen erschwert werden können.

Der Ansatz DEAL soll diesem Dilemma in Situationen parallel lernender Agenten entgegenwirken. Die Grundidee dabei ist, Vereinbarungen (engl. *Agreements*) zwischen Agenten mit unterschiedlichen Zielen zu ermöglichen und somit potentielle Konflikte, um beispielsweise gemeinsam genutzte Ressourcen, aufzulösen. Durch den Abschluss eines Vertrags zwischen Agent i und Agent j wird sichergestellt, dass i und j sich für die Dauer des Vertrags an bestimmte Verhaltensvorschriften halten, d.h. nur bestimmte, durch den Vertrag geregelte Aktionen, ausführen. Der Agent, dem durch einen solchen Vertrag eine höhere Belohnung entsteht, kann den Agenten, der sich durch

den Vertrag womöglich schlechter stellt, durch die Zahlung einer Kompensation entschädigen. Dadurch können sich beide Vertragspartner besserstellen als in der Ausgangssituation.

5.2.2 Verwandte Arbeiten

In diesem Abschnitt wird die Perspektive der kooperativen Spieltheorie auf Koalitionsbildungen zwischen selbstinteressierten (rationalen) Agenten dargestellt. Die kooperative Spieltheorie ist ein Zweig der Spieltheorie, der es den Agenten, im Gegensatz zur nicht kooperativen Spieltheorie, erlaubt, miteinander zu kommunizieren und verbindliche Vereinbarungen zu treffen [17]. Darüber hinaus werden Arbeiten aus dem Bereich des Multiagent Reinforcement Learning beschrieben.

5.2.2.0.1 Kooperative Spieltheorie Aus spieltheoretischer Sicht werden Koalitionsbildungsprozesse in drei Hauptaspekte unterteilt: Erstens, die eigentliche Aufgabe der Koalitionsbildung, bei der die Agenten herausfinden müssen, welche Koalitionen von Untergruppen von Agenten, gebildet werden können. Zweitens müssen die Agenten abschätzen, welche Auszahlungen sich aus verschiedenen Koalitionen ergeben. Drittens müssen sich die Agenten darauf einigen, wie sie die sich ergebenden Koalitionsgewinne untereinander aufteilen.

Um potenzielle Koalitionen zwischen rationalen Akteuren vorherzusagen, verwendet die kooperative Spieltheorie Lösungskonzepte wie den *Core* (für die Definition des Core siehe beispielsweise [17]), der darauf abzielt, stabile Lösungen zu finden, damit die Akteure keinen Anreiz haben eine bestehende Koalition zu verlassen. Andere Konzepte wie der *Shapley-Wert* [81] zielen darauf ab, eine Aufteilung der Koalitionsauszahlung zu finden, die den Beitrag jedes Agenten innerhalb der Koalition verwendet, um zu bestimmen, wie die Gesamtbelohnung aufgeteilt werden soll. Es gibt auch konkrete spieltheoretische Ansätze zum Finden von Koalitionen: Kraus [42] schlägt vor, dass Agenten Teile ihrer Aufgaben an andere Agenten, durch Verträge an andere Agenten, untervergeben können, im Austausch gegen eine versprochene zukünftige Belohnung, die entweder in Form von zukünftiger Hilfe oder als monetäre Entschädigung erfolgen kann. Gleichgewichte können dafür durch die Lösung eines Maximierungsproblems gefunden werden, für den Fall, dass keine Unsicherheit herrscht, also alle relevanten Informationen über die Umwelt bekannt sind. Ein weiterer spieltheoretischer Ansatz wird in [97] vorgestellt. Er beruht auf der Spezifikation eines Protokolls, das automatisierte Verhandlungen für verteilte Systeme der künstlichen Intelligenz (DAI) ermöglicht, bei denen Agenten potenziell von der Aufteilung ihrer Aufgaben profitieren können. In der Spieltheorie werden oft starke Annahmen getroffen, die im Hinblick auf die Verwendung in realen Anwendungen verschiedene Schwierigkeiten bringen:

- In der Spieltheorie werden Interaktionen zwischen mehreren Agenten üblicherweise als One-Shot-Spiele dargestellt, bei denen die Akteure gleich-

zeitig Entscheidungen treffen, die sofort zu Auszahlungen führen. Wie bereits in [45] erörtert, enthalten Multiagentensysteme in der realen Welt auch zeitliche Aspekte, so dass die Koalitionsbildung einen dynamischen Prozess darstellt, bei dem Koalitionen bei jedem Schritt gebildet und aufgelöst werden können und eine variable Dauer haben.

- In komplexen Szenarien ist es unrealistisch, a priori exakte Kenntnisse der Koalitionsauszahlungen zu haben, d.h. bevor eine solche Koalition überhaupt gebildet wurde. Außerdem kann der Koalitionswert von der Dauer der Koalition abhängen. Unter diesen Umständen ist es nicht klar, wie Lösungskonzepte wie der *Shapely Value*, die auf der Kenntnis der Koalitionsauszahlungen beruhen, angewendet werden können.

- Die Spieltheorie geht im Allgemeinen von individueller Rationalität bei der Entscheidungsfindung aus, was voraussetzt, dass alle Teilnehmer in der Lage sind, alle verfügbaren Informationen gleich effizient zu verarbeiten. Diese Annahme vernachlässigt individuelle Fähigkeiten und Verzerrungen bei der Entscheidungsfindung.

Der in diesem Kapitel vorgestellte Ansatz zur Bildung bilateraler Koalitionen zwischen eigennützigen Akteuren, basiert daher auf bilateralen Verträgen zwischen Agenten, wobei Koalitionen zu jedem Zeitschritt geschlossen werden können. Ein Vertrag zwischen zwei Agenten legt fest, welches Verhalten die beiden Vertragspartner befolgen müssen und wie die daraus resultierende gemeinsame Belohnung zwischen den Partnern verteilt wird. Um ohne Vorwissen über die Domänenaufgabe, den Wert von Koalitionen oder das Verhalten der gegnerischen Agenten zu lernen, werden die Wertefunktionen der Agenten verwendet, die mittels Reinforcement Learning gelernt werden. Die Unterschiede im Hinblick auf spieltheoretische Ansätzen sind im Speziellen:

- Anstatt von perfekter Rationalität der Akteure auszugehen, wird *deep* Reinforcement Learning verwendet, um Lernen zu ermöglichen. Die Optimalität von Entscheidungen wird somit durch iteratives Ausprobieren ermittelt und hängt durch das parallele Lernen aller Agenten auch von den aktuellen Strategien aller Agenten ab.

- Die Menge der möglichen Koalitionen wird dadurch begrenzt, dass jeweils nur zwei Agenten einen Vertrag miteinander abschließen können. Es ist jedoch möglich, dass ein Agent mehrere Verträge mit unterschiedlichen Vertragspartnern parallel führt. Die Agenten lernen autonom welche Koalitionen zu welchem Zeitpunkt abgeschlossen werden sollen.

- Da kein Vorwissen über den Wert potentieller Koalitionen vorausgesetzt wird, werden die Koalitionsauszahlungen aus den Q-Funktionen der Agenten abgeleitet, die im Voraus durch unabhängiges Training gelernt

werden. Die während des unabhängigen Trainings erlernten Wertefunktionen werden dann während des Trainings mit Verträgen verwendet, um die Vertragskompensationen abzuleiten, die ein Agent erhalten muss, um einem Vertrag beizutreten.

5.2.2.0.2 Koalitionen durch Reinforcement Learning Ein Ansatz zur Koalitionsbildung im Bereich des Reinforcement Learning wird in [16] vorgestellt, wobei ein Bayes'scher Ansatz verwendet wird, bei dem Koalitionswerte durch Schlussfolgerungen über Typen innerhalb einer Gruppe von potenziell heterogenen Agenten abgeleitet werden. Zu diesem Zweck treffen die Agenten Annahmen (engl. *Beliefs*) über andere Agenten, die durch wiederholte Interaktion aktualisiert werden. Basierend auf der Formulierung der sequentiellen sozialen Dilemmas (SSD) [45] wird in [91] ein Kooperations-Erkennungsnetzwerk trainiert, um den aktuellen Kooperationsgrad des Gegners zu identifizieren. Ausgehend von dem Verhalten des Gegners, kann ein Agent dann seinen eigenen Kooperationsgrad als beste Antwort wählen. Die Autoren Hughes et al. betrachten den Effekt von sozialen Präferenzen zur Förderung der Kooperation in verschiedenen Arten von SSDs, wobei gezeigt wird, dass Agenten, die ungleichheitsavers sind, die zeitliche Kreditvergabe verbessern können [35]. Beim Lernen mit gegnerischem Lernbewusstsein (LOLA) [27], bekommen Agenten Feedback über das Lernverhaltens anderer Agenten, durch einen zusätzlichen Term innerhalb der eigenen Lernregel. Es wird gezeigt, dass das Aufeinandertreffen zweier solcher Agenten zu Tit-for-Tat führen kann, einer Strategie, die für ihre Kooperationsbereitschaft und Robustheit gegenüber Ausbeutung durch unkooperative Agenten bekannt ist [7].

5.2.3 Konzept

Im Folgenden wird der Ansatz Distributed Emergent Agreement Learning (DEAL) vorgestellt. Grundlage, um die Agenten zu befähigen Verträge miteinander abzuschließen, ist, ähnlich wie bei den konditionalen bzw. bedingungslosen Marktmechanismen, eine Erweiterung der elementaren Aktionsräume der Agenten durch die Vertragsaktionen: $\{\textit{offer}, \textit{accept}\}$. Agenten können anderen Agenten also einen Vertrag anbieten (*offer*) und angebotene Verträge annehmen (*accept*). Zwei Agenten i und j schließen im Zeitschritt t einen Vertrag miteinander, wenn Agent i die Aktion *offer* und Agent j gleichzeitig die Aktion *accept* ausführt. Dabei ist im Speziellen wieder festzulegen, ob Vertragsangebote- und Annahmen wieder auf Agentenebene abgegeben bzw. angenommen werden (also i richtet an j und j akzeptiert von i) oder ob Angebote und Annahmen als globale Aktionen verstanden werden. Diese Entscheidung beeinträchtigt das Wachstum der Aktionsräume (vgl. Abschnitte 4.3.3 und 4.2.5.1). Durch den Abschluss eines Vertrags verpflichten sich beide Agenten für eine feste Laufzeit, den durch den Vertrag vorgeschriebenen Verhaltensvorschriften zu folgen.

Da die Agenten theoretisch durch die Teilnahme an einem Vertrag in Summe eine höhere Belohnung erzielen könnten, als es den einzelnen Agenten individuell möglich wäre, stellt sich die Frage, wie der Belohnungsüberschuss zwischen den beteiligten Agenten aufgeteilt wird. Eine Möglichkeit zur Aufteilung der Teambelohnung aus dem Bereich der kooperativen Spieltheorie ist der *Shapley-Value* [81]. Für die hier betrachteten Szenarien ist die Anwendung dieses Konzepts jedoch nicht möglich, da für die Verwendung des *Shapley-Value* alle möglichen Belohnungen aller Teilnehmer bekannt sein müssten. Der Ansatz, der hier beschrieben wird, beruht daher auf der Verwendung von bereits erlernten Werte-Funktionen. Vereinfacht ausgedrückt wird dabei die Kompensation, die ein Agent erhalten muss, damit er indifferent ist, zwischen einer optimalen Aktion und einer anderen, möglicherweise nicht optimalen Aktion, anhand der Differenz der Wertefunktionen für diese Aktionen ermittelt.

Das Konzept der wertebasierten Kompensationen wird im nächsten Abschnitt motiviert und theoretisch fundiert. Anschließend wird der Algorithmus *Distributed Emergent Agreement Learning* (DEAL) (dt. Verteiltes Lernen Emergenter Verträge) vorgestellt. Es folgt die Evaluation von DEAL zuerst für den Fall zweier Agenten und anschließend in einem erweiterten Szenario, mit bis zu 16 unabhängigen Agenten.

5.2.3.1 Wertebasierte Kompensationen

In diesem Abschnitt wird das Konzept wertebasierter Kompensationen vorgestellt, das in [78] erstveröffentlicht wurde. Wertebasierte Kompensationen sind durch die folgende Idee motiviert: Um Kooperation zwischen eigennützigen Agenten zu ermöglichen, muss der Agent, der während einer Kooperation individuell suboptimal handelt, indifferent gestimmt werden bezüglich der suboptimalen Aktion, die er im Zuge des Vertrags wählt und der individuell optimalen Aktion, auf die er zugunsten des Vertrags verzichtet. Läuft eine Kooperation für mehrere Zeitschritte, so muss der Agent für jeden Zeitschritt, für den er auf seine optimale Aktion verzichtet, kompensiert werden. Um indifferent zwischen einer optimalen Aktion a^* und einer suboptimalen Aktion a' im Zustand s_t zum Zeitpunkt t zu sein, muss ein Agent, der seinen erwarteten diskontierten Erlös maximiert, eine Kompensation erhalten, die den Unterschied in dem erwarteten diskontierten Erlös zwischen der Wahl von a' gegenüber a^* ausgleicht, wenn er anschließend wieder seiner optimalen Policy folgen kann. Der erwartete, diskontierte Erlös (genannt Wert) einer Handlung a im Zustand s unter der Policy π wird mit $q_\pi(s, a)$ bezeichnet und ist formal definiert durch [85] (vgl. Abschnitt 2.2.2):

$$q_\pi(s, a) = \mathbb{E}\big[\sum_{k=0}^{\infty} \gamma^k R_{k+1} | S_t = s, A_t = a\big] \tag{5.1}$$

Um einen Agenten indifferent zu halten bezüglich der Wahl einer suboptimalen Aktion a'_t und einer optimalen Aktion a^*_t zum Zeitpunkt t und im Zustand

s_t, muss gelten, dass die erwarteten und diskontierten Erlöse beider Aktionen gleich sind:

$$q_\pi(s_t, a_t^*) = q_\pi(s_t, a_t') \tag{5.2}$$

Dabei wird davon ausgegangen, dass im Anschluss wieder der Policy π gefolgt wird. Die Differenz zwischen $q_\pi(s_t, a_t^*)$ und $q_\pi(s_t, a_t')$ sei die erforderliche Kompensation c_t, also der Betrag, der dem Agenten zuteil werden muss, um bezüglich der Aktionen $a_t^{'}$ und a_t^* indifferent zu sein:

$$c_t \doteq q_\pi(s_t, a_t^*) - q_\pi(s_t, a_t') \tag{5.3}$$

Wird die Kompensation c_t unmittelbar, d.h. zum Zeitschritt t gezahlt, so ist der Agent echt indifferent bezüglich den Aktionen a_t^* und a'_t, da er in beiden Fällen den gleichen Ertrag erwarten kann. Wird die Kompensation zu einem späteren Zeitpunkt $t + n$ ausgezahlt, dann muss sie für jeden Zeitschritt n mit dem entsprechenden Diskontierungsfaktor γ aufgezinst werden, damit die Gleichung 5.2 weiterhin gilt. D.h., wenn die im Zeitschritt t entstandene Kompensation c_t im Zeitschritt $t + n$ gezahlt wird, so gilt für die zu zahlende aufgezinste Kompensation c_{t+n} zum Zeitpunkt $t + n$:

$$c_{t+n} = \frac{c_t}{\gamma^n} \tag{5.4}$$

Wird die Kompensation c_t durch die Zahlung von Teilbeträgen über mehrere Zeitschritte bezahlt, so muss jeweils nur der Rest der Gesamtkompensation aufgezinst werden. Sei $P = \{p_0, p_1, ..., p_n\}$ eine Folge von n Teilzahlungen, so dass P die Kompensation c_0 ausgleicht, dann muss jede Zahlung $p \in P$ in Bezug auf den Zeitschritt, zu dem sie gegeben wird, aufgezinst werden, d.h. die Gesamtkompensation c_{t+n} ergibt sich aus:

$$c_{t+n} = \frac{p_0}{\gamma^0} + \frac{p_1}{\gamma^1} + ... + \frac{p_n}{\gamma^n} = \sum_{k=0}^{n} \frac{p_k}{\gamma^k} \tag{5.5}$$

Da in dieser Arbeit lediglich episodische Aufgaben betrachtet werden, beschränken sich die möglichen Zahlungen auf den Verlauf einer Episode, d.h. Schulden können nicht über mehrere Episoden hinweg getilgt werden. Kompensationen, die am Ende einer Episode nicht ausbezahlt wurden, verfallen daher und werden nicht mehr berücksichtigt.

5.2.3.2 Verteiltes Lernen Emergenter Verträge

Nachdem im vorherigen Abschnitt wertebasierte Kompensationen eingeführt wurden, folgt in diesem Abschnitt die Beschreibung des Algorithmus *Distributed Emergent Agreement Learning* (*DEAL*), der auf Grundlage wertebasierter Kompensationen arbeitet. Ziel des Verfahrens ist es, Agenten autonom lernen zu lassen, in welchen Zuständen sie vom Eingehen eines Vertrags mit einem an-

deren Agenten profitieren. Zu diesem Zweck erfolgt das Lernen in zwei Phasen: Zunächst lernen die Agenten, wie sie mit der Umwelt interagieren können, ohne die Möglichkeit, Verträge miteinander zu schließen. Durch diese erste Trainingsphase lernen die Agenten unabhängige Wertefunktionen Q^i für $i \in \mathcal{N}$. Die auf diese Art gelernten Q-Funktionen werden dann persistiert und in der zweiten Trainingsphase eingesetzt.

In der zweiten Trainingsphase interagieren die Agenten in der gleichen Umgebung wie in der ersten Phase, mit folgendem Unterschied: In der zweiten Phase erhalten die Agenten über ihre erweiterten Aktionsräume die Möglichkeit Verträge miteinander zu schließen. Durch die Wahl einer Aktion aus dem kombinierten Aktionsraum, kann ein Agent nun seine Bereitschaft mitteilen, einen Vertrag mit einem anderen Agenten einzugehen. Die Wertefunktionen, die in der ersten Trainingsphase gelernt wurden, werden dann zur Berechnung der wertebasierten Kompensationen verwendet, sowie in Abschnitt 5.2.3.1 beschrieben. Darüber hinaus werden die in Phase 1 gelernten Q-Funktionen als Vertragspolicies genutzt, indem die Agenten, die einen Vertrag miteinander schließen ihre Aktionen *greedy* oder *non-greedy*, gemäß der gelernten Q-Funktionen wählen.

Algorithm 5 Distributed Emergent Agreement Learning

procedure DEAL
  $\mathbf{Q} \leftarrow Q^i$ for i $\in \mathcal{N}$ ▷ Vortrainierte Q-Funktionen
  for episode $e \in E$ **do**
    $A \leftarrow 0_{|\mathcal{N}|\times|\mathcal{N}|}$ ▷ Vertragsmatrix
    $C \leftarrow 0_{|\mathcal{N}|\times|\mathcal{N}|}$ ▷ Kompensationsmatrix
    for step $t \in T$ **do**
      for agent $i \in N$ **do**
        $a_t^i \sim \pi^i(s_t|\theta^i)$ ▷ Aktionswahl
      $A \leftarrow \text{getAgreements}(A, s_t, \mathbf{a}_t)$
      $\mathbf{a_t} \leftarrow \text{getAgreementActions}(\mathbf{Q}, A, C, s_t, \mathbf{a}_t)$
      $s_{t+1}, \mathbf{r_t} \leftarrow \text{step}(\mathbf{a_t})$
      $\mathbf{r_t} \leftarrow \texttt{compensate}(C, \mathbf{r}_t)$
      for agent $i \in N$ **do**
        save $(s_t, a_t^i, s_{t+1}, r_t^i)$ and train θ^i

Der kombinierte Aktionsraum des Agenten i enthält die Vertragsaktionen *no-op*, *offer*, *accept*, so dass der neue Aktionsraum $\mathcal{A}^i_{\text{DEAL}}$ als das kartesische Produkt zwischen dem ursprünglichen Aktionsraum $\mathcal{A}^i$ und den Vertragsaktionen gegeben ist: $\mathcal{A}^i_{\text{DEAL}} = \mathcal{A}_i \times \{\text{no-op}, \text{offer}, \text{accept}\}$. Alle laufenden Verträge einer Episode werden in einer Matrix $A \in \mathbb{N}^{|\mathcal{N}|\times|\mathcal{N}|}$ gespeichert, die zu Beginn jeder Episode mit Nullen initialisiert wird. Wann immer ein Vertragsangebot (*offer*) von einem anderen Agenten akzeptiert wird (*accept*), treten die beiden Agenten in eine Vertragsunterroutine ein, in der die Aktionen beider Agenten innerhalb des Vertrags aus den Vertragspolicies geholt werden. Dabei

werden die Aktionen für den Agenten, der das Vertragsangebot gemacht hat gemäß der Policy π^{offer} gewählt und für den akzeptierenden Agenten gemäß der Policy π^{accept}. Der Ablauf dazu ist in Algorithmus 6 beschrieben. In dieser Unterroutine werden die gelernten Q-Werte (mit **Q** bezeichnet in Algorithmus 5) aus der ersten Phase für die Aktionsauswahl verwendet: Während der anbietende Agent weiterhin seiner optimalen Policy folgt (z.B. basierend auf der argmax-Operation über die in der ersten Phase gelernten Q-Werte), muss der akzeptierende Agent einer nicht-optimalen Strategie folgen, indem er die Aktion mit dem niedrigsten zugehörigen Q-Wert wählt (argmin-Operation über die in der ersten Phase gelernten Q-Werte). Da diese Aktionsauswahl durch das Vertragsunterprogramm erzwungen wird, gibt es keine Möglichkeit für einen Agenten, die einmal getroffene Vereinbarung einseitig zu verletzen.

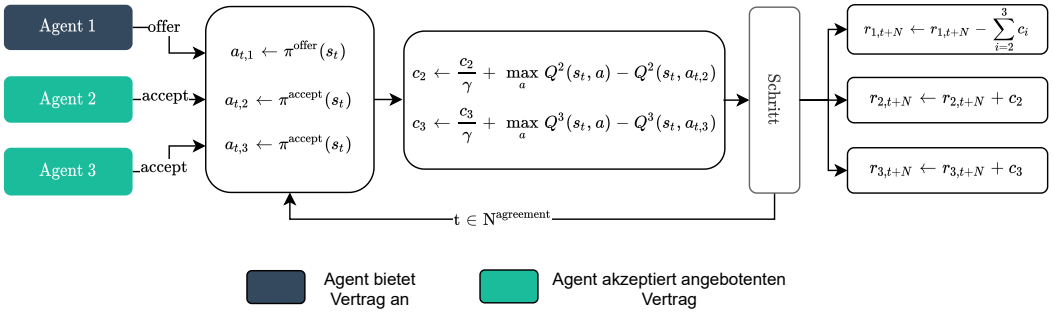


Abbildung 5.1: DEAL-Ablauf: Zwei Agenten (2 und 3) schließen bilaterale Verträge mit Agent 1, der die Agenten 2 und 3, für deren entgangene erwartete Belohnungen, entschädigen muss.

Zu beachten ist, dass Agenten gleichzeitig mehrere Verträge halten können, solange sie jeweils nur als Anbieter (durch die Aktion *offer*) oder als Akzeptor (durch die Aktion *accept*) auftreten. Das ergibt sich aus der Tatsache, dass der anbietende Agent, unabhängig vom zugrundeliegenden Vertrag, immer der Policy π^{offer} folgt und der akzeptierende Agent immer der Policy π^{accept} folgt. Da die Wahl der Aktionen vertragsübergreifend konsistent ist, ist es auch möglich mehrere Verträge mit unterschiedlichen Agenten zu halten. Es ist allerdings nicht möglich einen Vertrag anzubieten und selbst einen anderen Vertrag zu akzeptieren, da dann sowohl optimale als auch suboptimale Aktionen ausgeführt werden müssten. Da der akzeptierende Agent i in jedem Zeitschritt t der Vereinbarung eine Aktion mit niedrigeren Q-Wert wählen muss als unter seiner optimalen Policy, wird die Entschädigung durch die Differenz zwischen dem maximalen Q-Wert und dem Q-Wert der gewählten Aktion für den Zustand s_t berechnet, d. h. für die Aktionswahl mit minimalem Q-Wert: $c_t = \max_a Q^i(s_t, a) - \min_a Q^i(s_t, a)$. Die gemäß der Diskontierung aufgezinsten Kompensationen werden innerhalb einer Episode akkumuliert und in der Matrix $C \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ gespeichert, bis sie entweder vom anbietenden Agenten bezahlt werden oder bis die Episode endet. Nachdem die Umwelt in einen neuen Zustand s_{t+1} übergegangen ist, werden die Belohnungen r_t mit den fälligen

Kompensationen C verrechnet.

Abbildung 5.1 zeigt die schematische Vertragsroutine für den Fall dreier Agenten, wobei Agent 1 durch die Aktion *offer* jeweils ein Angebot an Agent 2 und 3 gemacht hat. Für die Laufzeit der Verträge werden nun die Aktionen aller Agenten gemäß den Vertragspolicies π^{offer} bzw. π^{accept} gewählt. Für jeden Zeitschritt des Vertrags wird die Kompensation anhand der jeweiligen Q-Werte für die gewählten Aktionen ermittelt und aufsummiert. Nach Vertragsende zahlt Agent 1 die Kompensationen an Agent 2 und 3, wodurch sich seine eigene Belohnung entsprechend vermindert.

Algorithm 6 Get Agreement Actions

procedure GETAGREEMENTACTIONS($\mathbf{Q}, A, C, s_t, \mathbf{a}_t$)
  for agreement a in A **do**
    get indices i, j for agents in a
    $\mathbf{a}'_{t,i} \leftarrow \text{argmin}_{a'}\, \mathbf{Q}_i(s_t, a')$
    $\mathbf{a}'_{t,j} \leftarrow \text{argmax}_{a'}\, \mathbf{Q}_j(s_t, a')$
    $C_{a,b} \leftarrow \mathbf{Q}_i(s_t, a'_{t,i}) - \mathbf{Q}_i(s_t, a_{t,i})$ ▷ in-place update
  return $\mathbf{a}'_t$

Die Verhaltensvorschriften π^{offer} und π^{accept} eines Vertrags definieren die exakten Aktionen, die während der Laufzeit des Vertrags von den Vertragspartnern ausgeführt werden. Im Allgemeinen könnten dafür auch andere Policies verwendet werden. Da in dieser Arbeit Szenarien betrachtet werden, in denen Agenten um verfügbare Ressourcen konkurrieren, lässt sich die Auswahl der Policies jedoch limitieren. Aufgabe des Vertrags ist es, Verhaltensvorschriften zu definieren, die dazu beitragen, den Konflikt zwischen den beiden Agenten zu koordinieren. Der Konflikt um Ressourcen lässt sich dadurch lösen, dass ein Agent seiner bereits gelernten Verhaltensvorschrift weiter folgen darf, d.h. er wählt Aktionen, beispielsweise *gierig* gemäß seiner aktuellen Wertefunktion. Der andere Agent hingegen verzichtet auf sein Recht seine besten Aktionen zu wählen und wählt Aktionen daher *nicht-gierig*.

5.2.3.2.1 Aktionsraum Bei direkter Implementierung des beschriebenen Mechanismus würde jeder Akteur angeben, an wen er ein Vertragsangebot machen möchte und von wem er bereit ist ein Vertragsangebot anzunehmen. Dadurch wächst der Aktionsraum exponentiell mit der Anzahl der Akteure. Wie bereits an den Beispielen der konditionalen und bedingungslosen Märkte gesehen, lässt sich das Wachstum des Aktionsraum effektiv begrenzen. Im Fall von DEAL lässt sich das durch folgende Beobachtung motivieren: Agenten, die bereit sind einen Vertrag zu akzeptieren, können mit allen möglichen Anbietern von Verträgen zusammengebracht werden, da ein akzeptierender Agent mehrere Verträge parallel abschließen kann, da er in jedem Vertrag der gleichen Policy π^{accept} folgen muss und die gleiche Kompensation erwarten kann. Für einen anbietenden Agenten kann es jedoch relevant sein, welchem Agenten

er einen Vertrag anbietet, da er vielleicht nur mit einem Agenten in räumlicher Nähe um Ressourcen konkurriert. Das Angebot eines Vertrags sollte daher an einzelne Agenten gerichtet werden können. Berücksichtigt man diese Annahmen, so wächst der Aktionsraum mit DEAL nur linear in der Anzahl der Agenten $|\mathcal{A}_{\text{DEAL}}| = |\mathcal{A}| * (|\mathcal{N}| + 1)$. Bei vier Agenten und vier möglichen Aktionen ($\{\text{left}, \text{right}, \text{up}, \text{down}\}$) beträgt die Mächtigkeit des Aktionsraums $|\mathcal{A}_{\text{DEAL}}| = |\{\text{left}, \text{right}, \text{up}, \text{down}\}| * |\{\text{no-op}, \textit{offer}_1, \textit{offer}_2, \textit{offer}_3, \textit{accept}\}| = 20$.

5.2.4 Evaluation

Die Evaluation von DEAL erfolgt in zwei Teilen. Im ersten Teil stehen Experimente mit zwei Agenten im Fokus, dabei wird insbesondere auch darauf eingegangen, wie sich verschiedene (externe) Parameter, wie beispielsweise die Ressourcenverfügbarkeit, auf Vertragsabschlüsse auswirken. Im zweiten Teil der Evaluation liegt der Fokus auf der Skalierbarkeit des Ansatzes, und es werden Experimente mit bis zu 16 unabhängig trainierten Agenten beschrieben. Für beide Teile der Evaluation wird eine Variante der Smartfactory verwendet, wie in Abschnitt 4.4.1 beschrieben.

5.2.4.1 Evaluation mit zwei Agenten

Dieser Teil der Evaluation beschränkt sich auf die Interaktion zweier Agenten in der Smartfactory-Umgebung. Dafür wird die Smartfactory folgendermaßen konfiguriert: Die Größe des Grids ist 5×5 und es stehen insgesamt drei Maschinen zur Verfügung. Jeder Agent erhält pro Episode einen Auftrag, für den drei Maschinen angefahren werden müssen. Es gibt zwei Maschinentypen, wobei ein Maschinentyp zweimal vorkommt und ein Maschinentyp nur einmal vorkommt. Die Aufträge der Agenten werden pro Episode zufällig aus den zwei möglichen Maschinentypen zusammengesetzt. Ein Auftrag kann entweder hohe oder niedrige Priorität haben. Hat ein Agent einen Auftrag mit hoher Priorität, so entstehen für jeden Zeitschritt, für den der Auftrag noch nicht abgearbeitet ist, Kosten in Höhe von c_{high} für den Agenten. Hat ein Agent einen Auftrag mit niedriger Priorität, so entstehen ihm Kosten in Höhe von c_{low}. Das Verhältnis von c_{high} zu c_{low} definiert die relative Priorität der Aufträge zueinander. Nachdem Maschinen von Agenten genutzt wurden, sind diese für eine gewisse Zeitdauer t_{inactive} nicht wiederverwendbar, d.h je größer t_{inactive} ist, desto größer auch der potentielle Konflikt zwischen den Agenten, der um die Nutzung der Maschinen entsteht.

5.2.4.1.1 Agenten und Training Für die Evaluation mit zwei Agenten werden beide Agenten in Form von Deep Q-Networks [56] repräsentiert, wobei die Architektur aus zwei neuronalen Netzen besteht, dem *Policy*-Netz und dem *Target*-Netz. Die beiden Netze bestehen jeweils aus zwei voll verknüpften Schichten, mit jeweils 32 Gewichten. Zur Aktivierung wird die Funktion *Exponential Linear Unit* (ELU) verwendet. Beide Agenten werden basierend

auf ihren partiellen Beobachtungen der Umgebung trainiert. Dazu hält jeder Agent sein eigenes *Experience Replay Memory*, das eine Kapazität von 20.000 Transitionen umfasst. Das *Target*-Netz wird alle 500 Schritte mit den Werten des *Policy*-Netzes gleichgesetzt. Die Lernrate beträgt $\alpha = 0,0005$ und zur Diskontierung wird $\gamma = 0,95$ verwendet. Jeder Trainingslauf besteht aus 4.000 Episoden in der Smartfactory, wobei die Episoden maximal 1000 Schritte dauern. Episoden können aber auch dadurch beendet werden, dass beide Agenten ihren Auftrag komplett abgearbeitet haben. Das Training der Netze beginnt nach 1.000 Simulationsschritten und die Aktualisierungen erfolgen auf der Basis einer Mini-Batchgröße von 64 Transitionen. Während des Trainings wird die Exploration durch die ϵ-greedy Aktionsauswahl, mit einem anfänglichen $\epsilon = 1$ und schrittweiser Dekrementierung von $0,000008$ bis zu einem Minimum von $\epsilon = 0,15$, stimuliert.

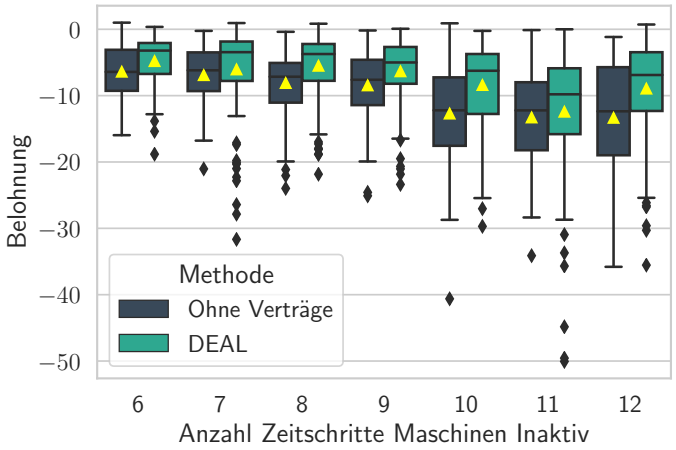
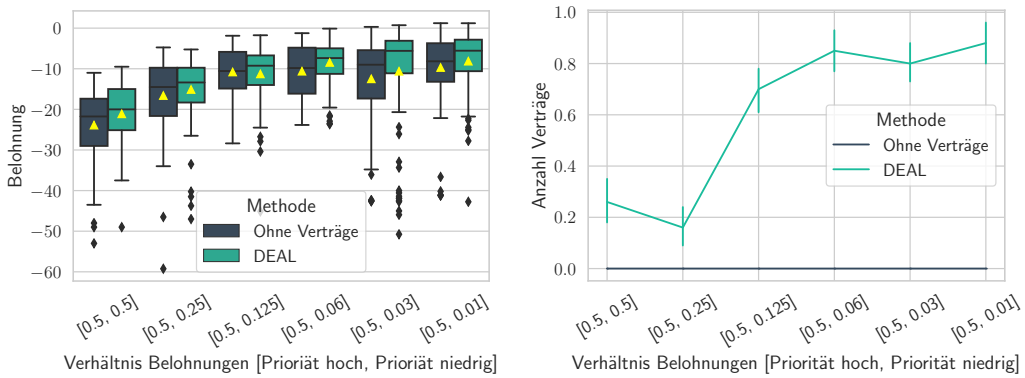


Abbildung 5.2: Vergleich der Gesamtbelohnung in der Smartfactory mit und ohne DEAL, für zwei Agenten und in Abhängigkeit der Ressourcenverfügbarkeit (Anzahl der Zeitschritte, nach der die Maschine nicht nutzbar ist).

5.2.4.1.2 Gesamtbelohnung Zuerst wird die episodische Gesamtbelohnung von *DEAL* betrachtet, d.h. die Summe der erzielten Belohnungen beider Agenten pro Episode. Aufgrund der höheren Kosten c_{high} pro Zeitschritt von Aufträgen mit hoher Priorität, im Vergleich zu den Kosten c_{high} von Aufträgen mit niedriger Priorität, misst die erzielte Gesamtbelohnung auch gleichzeitig den Grad der Koordination zwischen den Agenten, da hohe Belohnungen nur erreicht werden können, wenn Aufträge mit hoher Priorität schnell abgearbeitet werden. Abbildung 5.2 zeigt die die erzielte Gesamtbelohnung der Agenten mit und ohne DEAL, für unterschiedliche Werte der Maschinenverfügbarkeit t_{inactive}. Je höher t_{inactive} desto länger die Zeitspanne, während derer eine Maschine nach der Nutzung nicht verwendet werden kann, wodurch potentielle Engpässe in der Fabrik erhöht werden, der Konflikt zwischen den Agenten also

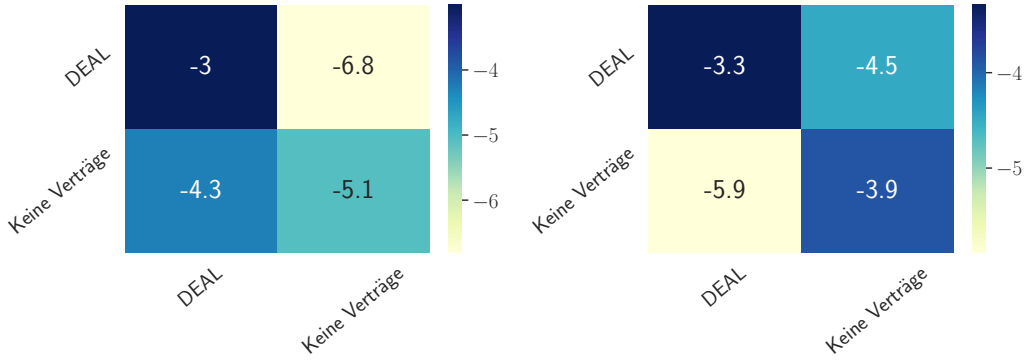


(a) Belohnung (b) Anzahl Verträge

Abbildung 5.3: Erzielte Belohnung und Anzahl abgeschlossener Verträge, in Abhängigkeit zu den Kosten für Aufträge mit hoher und niedriger Priorität.

größer wird. Die Boxplots in Abbildung 5.2 zeigen jeweils die erzielte Gesamtbelohnung von 100 Testepisoden, die nach Abschluss der 4.000 Trainingsepisoden simuliert wurden (Mittelwerte durch Dreiecke kenntlich gemacht). Für alle Werte der Maschinenverfügbarkeiten ist die durchschnittliche Belohnung im Fall von DEAL höher als DQN ohne DEAL. Auch weisen die DEAL-Ergebnisse in den meisten Fällen weniger Streuung auf. Die Unterschiede in den Verfahren scheinen für den untersuchten Bereich nicht direkt von t_{inactive} beeinträchtigt zu werden.

Neben dem Parameter t_{inactive} zur Steuerung der Maschinenverfügbarkeit, ist das Verhältnis der Kosten c_{high} zu c_{low} ausschlaggebend für die relative Priorität der Aufträge. Je höher die Kosten c_{high} sind, die pro Zeitschritt für die Abarbeitung eines Auftrags mit hoher Priorität entstehen, im Verhältnis zu den Kosten c_{low}, die während der Abarbeitung eines Auftrags mit niedriger Priorität entstehen, desto mehr kann durch ein koordiniertes Vorgehen der Agenten die Gesamtbelohnung verbessert werden. Es gilt: Je größer die Differenz, desto größer ist der potenzielle Nutzen eines koordinierten Vorgehens der Agenten. Wenn DEAL zur Koordination der Agenten beiträgt, so sollten in diesem Fall dann ebenfalls mehr DEAL-Verträge geschlossen werden. Zur Überprüfung dieser Hypothese wird das Verhältnis der Kosten c_{high} zu c_{low} variiert, mit $c_{\text{high}} = 0,5$ und $c_{\text{low}} \in [0.5, 0.25, 0.125, 0.06, 0.03, 0.01]$. Die Resultate sind in Abbildung 5.3 dargestellt. Hierbei zeigt sich, dass die Läufe mit DEAL in fünf von sechs Konfigurationen eine höhere durchschnittliche Gesamtbelohnung aufweisen (vgl. Abbildung 5.3). Interessanterweise zeigt DEAL selbst im Falle symmetrischer Kosten ($c_{\text{high}} = c_{\text{low}} = 0,5$) eine höhere Gesamtbelohnung als reines DQN ohne Verträge, was an der Nutzung der bereits gelernten Vertragspolicies liegen kann. Die durchschnittliche Anzahl der realisierten Verträge pro Episode steigt mit zunehmender Asymmetrie der Kosten (Abbildung 5.3b). Es gilt also, dass Verträge bei größerer Differenz von c_{high} zu c_{low} mit



(a) Agent 1 (b) Agent 2

Abbildung 5.4: Empirisch ermittelte Auszahlungsmatrizen für zwei Agenten mit und ohne DEAL.

höherer Wahrscheinlichkeit zustande kommen.

5.2.4.1.3 Dominanz der Strategie DEAL Stellt man sich zwei Agenten vor, die vor der Wahl stehen, ob sie für einen Trainingslauf DEAL verwenden sollen oder nicht, so ergibt sich daraus ein Normalform-Spiel mit zwei Aktionen. In diesem Abschnitt soll untersucht werden, ob es aus individueller Sicht rational ist DEAL zu verwenden bzw. welche Nash-Gleichgewichte es in diesem Meta-Spiel gibt. Diese Frage wird mit Methoden der empirischen Spieltheorie untersucht: Man stelle sich dazu vor, dass zwei Agenten vor Beginn jeder Episode entscheiden müssen, ob sie während des Trainings mit der Möglichkeit oder ohne die Möglichkeit Verträge nutzen zu können, trainieren möchten. Daraus ergibt sich ein Normalform-Spiel, das in Form einer 2×2-Matrix dargestellt werden kann. Um herauszufinden, welche Nash-Gleichgewichte es in diesem Spiel gibt, werden noch die Auszahlungen (Belohnungen) für die verschiedenen Spiel-Ausgänge benötigt. Da diese Auszahlungen zunächst unbekannt sind, werden sie hier durch Simulationen des Spiels ermittelt.

Um die Auszahlungsmatrizen für die vier verschiedenen Kombinationen des oben beschrieben Matrix-Spiels zu ermitteln, werden Simulationen durchgeführt, bei denen bereits fertig trainierte Agenten miteinander interagieren. Beiden Agenten wird dabei eine fixe Policy π^i zugewiesen, wobei π^i entweder eine vortrainierte DEAL-Policy π^{DEAL} ist oder eine vortrainierte Policy ohne Verträge $\pi^{\text{NO-DEAL}}$. Daraus ergeben sich insgesamt vier mögliche Kombinationen von Policies $(\pi^1, \pi^2) \in \{(\pi^{\text{DEAL}}, \pi^{\text{DEAL}}), (\pi^{\text{DEAL}}, \pi^{\text{NO-DEAL}}), (\pi^{\text{NO-DEAL}}, \pi^{\text{DEAL}}), (\pi^{\text{NO-DEAL}}, \pi^{\text{NO-DEAL}})\}$. Um nun die Auszahlungen für die Kombination dieser Policies zu ermitteln, werden jeweils 250 Episoden der beiden Agenten gespielt. Die daraus resultierenden Belohnungen für Agent 1 und 2 sind in Abbildung 5.4 dargestellt (die Belohnungen wurden über die 250 Episoden gemittelt).

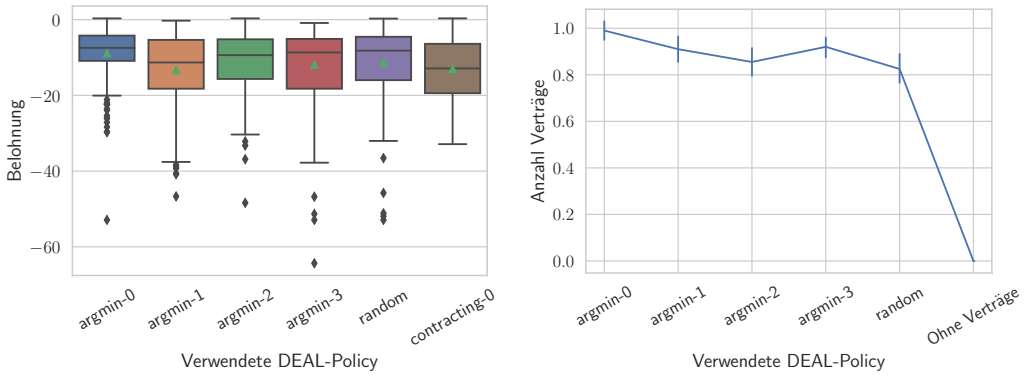
Wie in Abbildung 5.4 zu sehen, haben beide Agenten die höchste erwartete

Belohnung, wenn beide Agenten DEAL-Policies verwenden. Hier liegt die empirisch ermittelte Belohnung für Agent 1 bei -3 und für Agent 2 bei $-3,3$. In allen anderen Fällen sind die erzielten Belohnungen niedriger. Insgesamt weist das dargestellte Matrixspiel zwei Nash-Gleichgewichte in reinen Strategien auf: einmal wenn beide Agenten DEAL nutzen $(\pi^{\text{DEAL}}, \pi^{\text{DEAL}})$, das andere Gleichgewicht wenn beide Agenten DEAL nicht nutzen $(\pi^{\text{NO-DEAL}}, \pi^{\text{NO-DEAL}})$, da in beiden Fällen keine Anreize bestehen von diesen Strategien abzuweichen, wenn davon ausgegangen wird, dass der andere Agent auch nicht davon abweicht. Allerdings definiert das Nash-Gleichgewicht $(\pi^{\text{DEAL}}, \pi^{\text{DEAL}})$ ein optimales Gleichgewicht, da hier die Summe der Belohnung maximal ist. Es ist daher individuell rational Verträge zu nutzen, wenn davon ausgegangen werden kann, dass der gegnerische Akteur ebenfalls Verträge nutzt. Ist jedoch davon auszugehen, dass der Mitspieler keine Verträge nutzt, dann ist es individuell rational, ebenfalls keine Verträge einzusetzen.

5.2.4.1.4 Einfluss der Vertrags-Policy Bei der Evaluation des Ansatzes DEAL wurden bisher die Vertragspolicies $\pi^{\text{accept}} = \operatorname{argmin} Q_i(s,a)$ für einen akzeptierenden Agenten verwendet und $\pi^{\text{offer}} = \operatorname{argmax} Q_i(s,a)$ für einen Agenten, der einen Vertrag anbietet. Die konkrete Wahl der argmin-Operation für den Akzeptor, motiviert sich darüber, dass in den hier betrachteten Szenarien ein Konflikt um gemeinsam genutzte Ressourcen zwischen den Agenten besteht. Es ist deshalb plausibel anzunehmen, dass nur ein Agent seiner optimalen Policy *greedily*, d.h. gemäß der argmax-Operation, folgen kann.

Die Vertragspolicies könnten darüber hinaus auch andere Möglichkeiten berücksichtigen. Naheliegende Kandidaten dafür wären neben der argmin-Operation, die Policy, die die Aktion mit dem zweitniedrigsten Q-Wert anwendet (bezeichnet mit $\operatorname{argmin} -1$), dem drittniedrigsten Q-Wert (bezeichnet mit $\operatorname{argmin} -2$), dem viertniedrigsten Q-Wert (bezeichnet mit $\operatorname{argmin} -3$) und die Policy, die zufällig eine Aktion wählt (bezeichnet mit random). Zu diesen Varianten wurden die gleichen Experimente wie zuvor durchgeführt. Abbildung 5.5 zeigt die Ergebnisse für die beschriebenen Policy-Varianten. Dargestellt sind dabei die erzielte Gesamtbelohnung sowie die Gesamtanzahl realisierter Verträge, unter Verwendung verschiedener Policies. Es zeigt sich, dass die durchschnittliche Gesamtbelohnung für alle betrachteten Policies am höchsten im Falle der argmin-Operation ist. Die random-Policy schneidet durchschnittlich am schlechtesten ab. Darüber hinaus ist die durchschnittliche Anzahl abgeschlossener Verträge im Fall der argmin-Operation am höchsten (siehe 5.5b).

5.2.4.1.5 Interpretierbarkeit der Modelle In diesem Abschnitt wird die Interpretierbarkeit der trainierten Modelle untersucht. Die Frage der Interpretierbarkeit ist insbesondere im Bereich tiefer neuronaler Netze relevant, da aus der Betrachtung der trainierten Gewichte eines Netzes keine Einschätzung möglich ist, warum es zu bestimmten Vorhersagen bzw. Entscheidungen kommt. Eine Möglichkeit, herauszufinden wie bestimmte Neuronen auf bestimmte Eingangs-



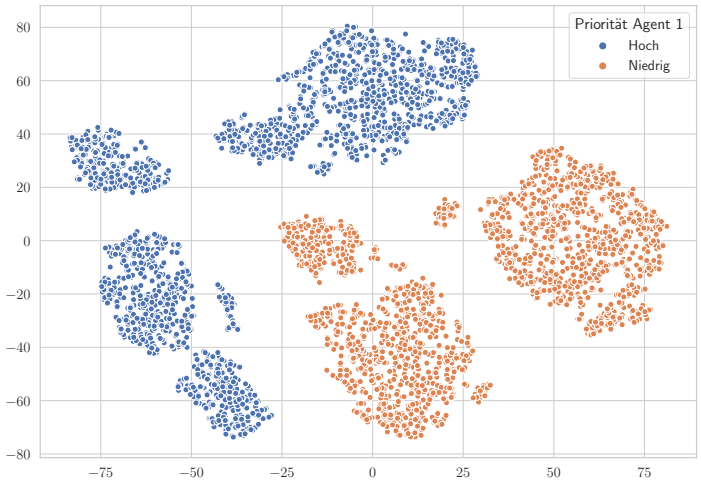
(a) Agent 1 (b) Agent 2

Abbildung 5.5: Ergebnisse unter Verwendung unterschiedlicher DEAL-Policies.

daten reagieren, liegt in der zweidimensionalen Visualisierung der Ausgangssignale der letzten verborgenen Schicht, mittels des *t-SNE*-Verfahrens [50], einer Visualisierungsmethode, die es ermöglicht hochdimensionale Daten im 2- oder 3-D Raum darzustellen.

Hier soll nun dargestellt werden, welche speziellen Spielzustände in der Smartfactory-Umgebung welchen DEAL-Entscheidungen zugeordnet werden. Dazu wurden 5.000 randomisierte Startzustände der Smartfactory erzeugt und dazu jeweils die Ausgangssignale der letzten verborgenen Schicht eines trainierten Agenten-Modells gespeichert. Der daraus entstehende Datensatz mit 5.000 Datenpunkten, wurde dann für die *t-SNE* Analyse verwendet. Die sich daraus ergebenden Punktewolken sind in den Abbildungen 5.6 dargestellt. Um einzelne Punkte nun zu bestimmten interpretierbaren Zuständen der Umgebung zuzuordnen, wurden die Datenpunkte mit verschiedenen Attributen annotiert, die über die Farbe kenntlich gemacht sind. Abbildung 5.6a zeigt die Zuordnung der Auftragsprioritäten zu den verschiedenen Startzuständen, und Abbildung 5.6b zeigt in welchen Startzuständen es im Lauf der Episode zu einem erfolgreichen Vertragsabschluss zwischen den Agenten kam.

Abbildung 5.6a lässt erkennen, dass das trainierte Netz deutlich zwischen Zuständen unterscheidet, die entweder hohe oder niedrige Priorität haben, da die Farben deutlich zu verschiedenen Clustern zugeteilt sind. Ein ähnliches Bild ergibt sich für die Zustände, die zu abgeschlossenen Verträgen führen (Abbildung 5.6b). Hier ist zwar eine leichte Durchmischung der Cluster zu erkennen, aber auch, dass der Agent insbesondere dann erfolgreich Verträge abschließt (als Anbieter), wenn er einen Auftrag mit hoher Priorität hat, da die entsprechenden Cluster nahezu identisch gefärbt sind. Zur Interpretierbarkeit der trainierten Modelle lässt sich sagen, dass es möglich ist bestimmte für den Mensch interpretierbare Zustände auch in den gelernten Repräsentationen der Modelle wiederzufinden, wie sich durch die starke Homogenität, bezogen auf die Farbe der Cluster, erkennen lässt.



(a) Zugeordnete Prioritäten.

(b) Erfolgreiche Vertragsabschlüsse im Episodenverlauf.

Abbildung 5.6: t-SNE-Einbettung der letzten verborgenen Schicht eines trainierten Agenten mit Clustern, die nach bestimmten Spielzuständen gefärbt sind, d. h. (a) nach den zugehörigen Prioritäten und (b) nach den erfolgreichen Verträgen, die von diesem Agenten, während einer einzelnen Episode, angeboten wurden.

5.2.4.2 Evaluation mit mehr als zwei Agenten

In diesem Abschnitt wird die Evaluation nun erweitert und umfasst bis zu 16 unabhängig trainierte Agenten. Im Gegensatz zu den zuvor beschrieben Experimenten mit zwei Agenten, soll dieser Abschnitt darstellen, inwiefern DEAL mit der Anzahl der Agenten skaliert. Darüber hinaus wird auch die Verteilung der Belohnung auf die Agenten untersucht sowie eine Robustheitsanalyse durchgeführt.

5.2.4.2.1 Evaluations-Umgebung Das zugrunde liegende Szenario bleibt auch für diesen Teil der Evaluation die in Abschnitt 4.4.1 vorgestellte Smartfactory. Im Vergleich zu den zuvor beschriebenen Experimenten mit zwei Agenten, wird ein Layout der Fabrik, wie in Abbildung 4.14 dargestellt, verwendet mit einer Gridgröße von 12×8. Die Anzahl der Maschinen wird auf vier erhöht, wobei jeweils zwei Maschinen vom gleichen Typ sind. Zwischen den Maschinen befinden sich teilweise Wände, so dass der Zugang nur über die freien Pfade möglich ist. Jeder Agent erhält zu Beginn einer Episode einen Auftrag, bestehend aus drei unterschiedlich anzufahrenden Maschinen, wobei die Aufträge zufällig generiert werden. Um den Wettbewerb zwischen den Agenten zu verschärfen, gibt es pro Episode genau einen Auftrag mit hoher Priorität. Der Agent mit diesem Auftrag erhält für die erfolgreiche Abarbeitung eine hohe Belohnung $r_{\text{high}} = 5$. Dazu ist es aber notwendig, dass er es schafft seinen Auftrag als erster abzuarbeiten. Für alle Agenten, die einen Auftrag mit niedriger Priorität haben gilt, dass sie eine niedrige Belohnung $r_{\text{low}} = 1$ erhalten, unabhängig vom Zeitpunkt der Abarbeitung. Eine Episode dauert maximal 200 Zeitschritte.

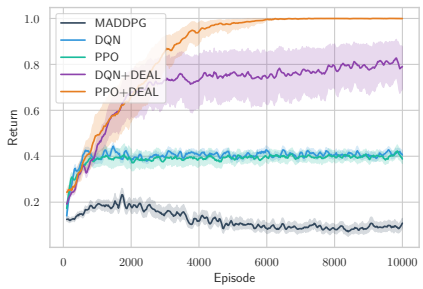
5.2.4.2.2 Agenten und Training Zur Evaluation von *DEAL* werden Agenten entweder als Deep Q-Networks (DQN) [56] oder durch den Algorithmus Proximal Policy Optimization (PPO) [80] repräsentiert. Alle Trainingsläufe bestehen aus 10.000 Episoden für bis zu acht Agenten bzw. 30.000 Episoden im Fall von 16 Agenten. Das *Policy*- und Target-Netz der DQN-Agenten besteht aus zwei voll verknüpften Schichten mit 32 bzw. 16 Gewichten und exponentiellen linearen Einheiten (ELUs) [20] als Aktivierungsfunktionen. Während des Trainings wird die Erfahrung in einem *Replay Memory*-Puffer E mit einer maximalen Kapazität von 20.000 Transitionen gespeichert. Das *Target*-Netz wird alle 500 Schritte aktualisiert und das Training beginnt nach anfänglichen 200 Simulationsschritten. Das Lernen erfolgt auf der Basis eines Mini-Batches der Größe 64, mit einer Lernrate $\alpha = 0,0005$ und Diskontierung $\gamma = 0,95$. Während des Trainings wird exploratives Verhalten durch die ϵ-greedy Aktionsauswahl gefördert, wobei ϵ linear auf ein Minimum von $\epsilon = 0,1$ agekühlt wird. PPO-Agenten bestehen aus einem *Actor*- und *Critic*-Netzwerk mit zwei versteckten Schichten der Größe 32 und tangens hyperbolicus (tanh) als Aktivierungsfunktion. Die Modelle werden alle 5.000 Schritte für vier Epochen mit einer Lernrate von $0,002$, trainiert.

Für die Wertefunktionen, die für das Training der DEAL-Agenten verwendet werden, um die Kompensationen für die Verträge zu berechnen, wurde das Q-Netz verwendet, das in der ersten Trainingsphase den höchsten durchschnittlichen Ertrag erzielt hat. Die Vertragsdauer wurde auf $N_{agreement} = 1$ gesetzt. Das Training von DEAL erfordert die doppelte Rechenkapazität im Vergleich zum unabhängigen Training, da die Trainingsepisoden und Episodenschritte für beide Trainingsphasen gleich sind und die neuronalen Netze die gleiche Architektur und Anzahl von Parametern haben. Während der zweiten Trainingsphase von DEAL ist kein zusätzliches Training der neuronalen Netze erforderlich, da nur Vorwärtsdurchläufe durchgeführt werden. Die Anzahl der Modelle, die in Trainingsphase 2 im Speicher gehalten werden müssen, verdoppelt sich im Vergleich zum Training ohne DEAL.

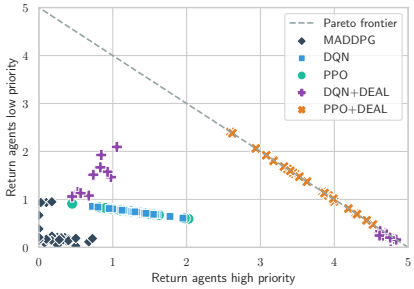
5.2.4.2.3 Ergebnisse Es werden fünf Arten von Agenten verglichen: DQN- und PPO-Agenten ohne DEAL, d.h. ohne die Möglichkeit Verträge zu schließen. DQN- und PPO-Agenten mit DEAL, also mit der Möglichkeit Verträge zu schließen, sowie zusätzlich Ergebnisse des Algorithmus *Multiagent Deep Deterministic Policy Gradient* (MADDPG) [48], um einen Vergleich zu einem zentralisierten Ansatz ohne unabhängiges Lernen herzustellen. Für alle Agenten-Typen wurden 30 unabhängige Trainings-Läufe durchgeführt, so dass alle Abbildungen auch Mittelwerte und Konfidenzintervalle, zum Konfidenzniveau von 95%, beinhalten. Zur Evaluation wurden Szenarien mit vier, acht und 16 Agenten in der Smartfactory untersucht.

Abbildung 5.7 zeigt die Ergebnisse für den Fall von vier, acht und 16 Agenten, wobei jeweils die (normalisierte) Gesamtbelohnung und die Verteilung der Gesamtbelohnung auf Agenten mit niedriger und hoher Priorität dargestellt ist. Das insgesamt höchste Gesamtergebnis wird am Ende des Trainings in allen Szenarien durch die Kombination von PPO+DEAL erzielt. DQN+DEAL erreicht nicht das gleiche Niveau wie PPO+DEAL, benötigt aber, zumindest für den Fall von 16 Agenten, weniger Trainingsepisoden, um auf die schlussendlich erreichte Leistung zu gelangen. Die Verfahren PPO und DQN ohne DEAL bleiben auf einem verhältnismäßig geringen Gesamtniveau zurück. Der Ansatz MADDPG verschlechtert sich im Laufe des Trainings sogar noch und schneidet insgesamt am schwächsten ab.

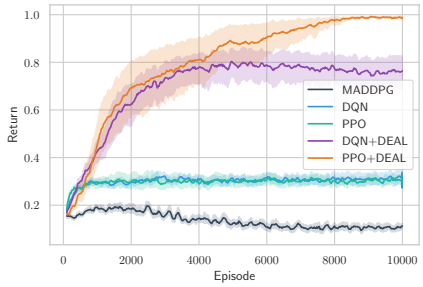
Neben der Höhe der erzielten Gesamtbelohnung stellt sich die Frage, wie sich die erzielte Gesamtbelohnung auf die einzelnen Agenten des Systems verteilt. Hier soll speziell untersucht werden, wie sich die Belohnung auf die Gruppe der Agenten mit Aufträgen niedriger Priorität und Agenten mit Aufträgen hoher Priorität verteilt, da Verträge ja primär darauf abzielen diesen Konflikt zu lösen. Die Verteilung der resultierenden Belohnungen für die jeweiligen Trainingsläufe ist in den Abbildungen 5.7b, 5.7d und 5.7f dargestellt. Dabei zeigt die x-Achse den erzielten Anteil an der Gesamtbelohnung für Agenten mit hoher Priorität, und auf der y-Achse ist der Anteil der Agenten mit niedriger Priorität abgetragen. Die gestrichelte Linie deutet die Pareto-Grenze an, d.h.



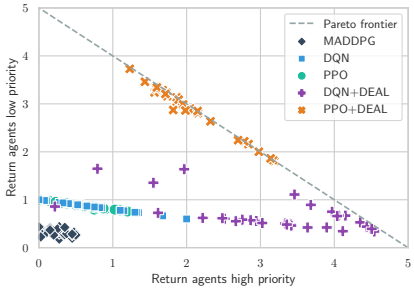
(a) Belohnung bei vier Agenten



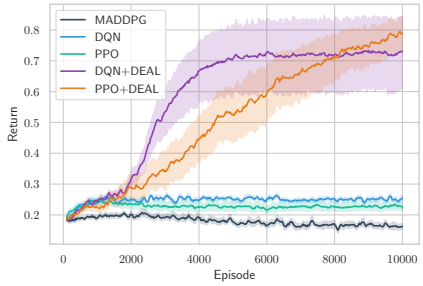
(b) Verteilung bei vier Agenten



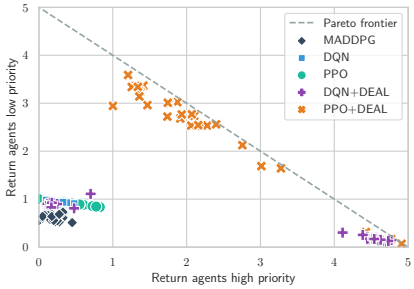
(c) Belohnung bei acht Agenten



(d) Verteilung bei acht Agenten



(e) Belohnung bei 16 Agenten



(f) Verteilung bei 16 Agenten

Abbildung 5.7: Erzielte Belohnung und Verteilung der Belohnung für vier, acht und 16 Agenten in der Smartfactory.

die Menge der Punkte die Pareto-optimal sind.

Generell liegen die Ergebnisse der *DEAL*-Agenten, insbesondere für PPO+DEAL, deutlich näher (oder auf) der Pareto-Grenze als die Verfahren ohne DEAL. Es zeigt sich, dass ein zunehmender Teil der erzielten Gesamtbelohnung mit wachsender Anzahl an Agenten an die Gruppe von Agenten mit niedriger Priorität geht. Dieses Ergebnis würde man von einem effektiven Marktansatz erwarten, da durch mehr Agenten im System auch die Gruppe der Agenten mit niedrigen Prioritäten im gleichen Maße wächst und somit immer mehr Agenten an der erzielten Gesamtbelohnung partizipieren sollten. Die Verteilung der Gesamtbelohnung ist aber maßgeblich von dem zugrundeliegenden Verfahren (PPO oder DQN) beeinflusst, denn die Ergebnisse von PPO+DEAL liegen insgesamt weiter links, weisen also generell einen höheren Anteil der Belohnung den Agenten mit niedriger Priorität zu.

5.2.4.2.4 Stabilität Bisher wurde davon ausgegangen, dass alle Agenten des Systems Verträge nutzen können und wollen. Für den Fall von zwei Agenten wurde bereits gezeigt, dass es aus Sicht eines einzelnen Agenten auch immer rational ist dies zu tun, da die erwarteten Belohnungen im Falle, dass beide Agenten Verträge nutzen, am höchsten sind. Es schließt sich die Fragen an, ob DEAL diese Stabilität auch für mehr als zwei Agenten aufweist. Um diese Fragen zu beantworten, wird ein Szenario mit insgesamt 16 Agenten untersucht. Es werden schrittweise Veränderungen der Anzahl der DEAL-Agenten an der Gesamtanzahl der Agenten vorgenommen von null DEAL-Agenten bis zu 16 DEAL-Agenten. Die Policy π_i des Agenten i wurde dazu entweder aus einer Menge von zehn vortrainierten DEAL-Policies Π^{DEAL} oder aus einer Menge von zehn vortrainierten Policies ohne *DEAL* $\Pi^{No\text{-}DEAL}$ gezogen. Für jede Zusammensetzung von DEAL-Agenten und Standard-Agenten wurden jeweils 1.000 Test-Episoden gespielt und die Ergebnisse dieser Episoden pro Agent gemittelt. Die Ergebnisse aller Konfigurationen sind in Abbildung 5.8 dargestellt, wobei die Zelle j, i den durchschnittlichen Ertrag des Agenten i bei einer Gesamtzahl von j DEAL-Agenten in der Simulation darstellt. Generell steigt die durchschnittliche erzielte individuelle Belohnung proportional mit der Anzahl der DEAL-Agenten in der Umgebung. Allerdings gibt es einzelne Agenten, die für bestimmte Zwischenkonfigurationen eine höhere erwartete Belohnung haben. Da die individuellen Belohnungen mit der Anzahl der DEAL-Agenten im System wachsen, ist davon auszugehen, dass es individuell ebenfalls rational ist DEAL zu nutzen. Dabei ähnelt das Szenario dem sozialen Dilemma *Stag-Hunt*: es ist für einzelne Agenten ratsam *DEAL* zu nutzen, solange davon ausgegangen werden kann, dass genügend andere Agenten ebenfalls *DEAL* einsetzen.

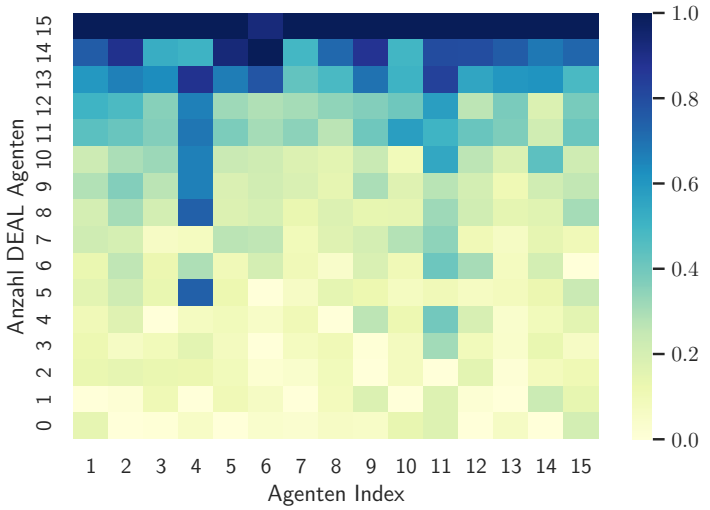


Abbildung 5.8: Entwicklung der individuellen Belohnungen nach Gesamtanzahl von DEAL-Agenten im System.

5.3 Kooperation in Sozialen Dilemmas

Die in den vorherigen Abschnitten vorgestellten Marktmechanismen gehen von der Möglichkeit einer *positiven* Incentivierung, durch die Umverteilung von Belohnungen zwischen den Agenten, aus. Die Agenten können Anreize geben, da sie unterschiedliche Belohnungen durch die Erreichung ihrer Ziele realisieren. Diese Möglichkeit besteht nicht in *zero-sum* Szenarien, doch auch in sozialen Dilemmas, die nicht *zero-sum* sind, wie dem Prisoner's Dilemma (vgl. Abschnitt 2.1.3) ist eine kompensationsbasierte Incentivierung nicht möglich: Die Belohnung, die ein Agent durch einseitiges Abweichen von beidseitiger Kooperation erhält, reicht nicht aus, um den anderen Agenten dafür zu kompensieren und ihn dadurch indifferent zu machen, da das globale Optimum in beidseitiger Kooperation liegt. In diesem Abschnitt werden die bisherigen Ansätze daher auf deren Anwendbarkeit in solchen Situationen untersucht. In verschiedenen sozialen Dilemmas mit zwei Spielern und dem N-Spieler Prisoner's Dilemma wird gezeigt, dass der konditionale Markt mit negativem Preis, den erreichten Grad an Kooperation erheblich steigern kann.

5.3.1 Idee und Motivation

Soziale Dilemmas sind allgegenwärtig im Bereich zwischenmenschlicher Interaktionen, insbesondere im Zusammenhang mit geteilten Ressourcen wie die Überfischung der Meere, Bewässerungssysteme oder die Abholzung von Wäldern (siehe Tragik der Allmende [32]) und daher zentraler Gegenstand der Spieltheorie. Durch die zunehmende Bedeutung autonomer KI-Systeme ist die Frage nach den Bedingungen für Kooperation auch in diesem Bereich rele-

vant geworden [21]. Soziale Dilemmas werden dabei verwendet, um lernende Algorithmen in Bezug auf deren Kooperationsfähigkeit zu evaluieren. Inwiefern auch marktbasierte Ansätze dazu beitragen können, Kooperation in sozialen Dilemmas zu unterstützen, wird daher in diesem Abschnitt behandelt. Stellt man sich die Situation des Prisoner's Dilemma vor, so wird klar, dass eine direkte Anwendung der in Kapitel 4 vorgestellten Ansätze unplausibel erscheint: Ausgehend von einer Situation, in der einer der Agenten kooperativ handelt und ein Agent nicht kooperativ handelt, gibt es keine Möglichkeit eine Pareto-Verbesserung durch einen Handel durchzuführen, da mindestens der nicht kooperative Agent durch eine andere Aktionswahl schlechter gestellt werden würde. Anstelle positiver Incentivierung wird der Ansatz daher in diesem Abschnitt zu Incentivierung durch Bestrafung verändert.

Die Idee, dass Spieler sich gegenseitig bestrafen können, wurde bereits früher sowohl theoretisch als auch in der experimentellen Forschung behandelt [63], wobei eine positive Auswirkung von Sanktionierungen auf die allgemeine Bereitschaft zur Kooperation festgestellt wurde. In früheren Arbeiten wurden Strafen jedoch als eine Operation definiert, die Kosten für diejenigen verursacht, die andere bestrafen. Dieser Umstand kann ein soziales Dilemma zweiter Ordnung aufwerfen [39]: Wenn der Akt der Bestrafung anderer Agenten mit Kosten verbunden ist, sei es durch die Zeit, die aufgewendet wird, um eine Strafe zu verhängen, die nicht zur Steigerung des eigenen Nutzens verwendet werden kann [66], oder die tatsächliche Auszahlung, die investiert werden muss, um jemanden zu bestrafen [63], dann könnten sich die Spieler entscheiden, die Kosten der Bestrafung anderen zu überlassen, d. h. sie werden zu Trittbrettfahrern.

Der Ansatz, der in diesem Abschnitt beschrieben wird, trägt dazu bei, beide Dilemmas parallel zu lösen:

- Indem man den Spielern ermöglicht, andere Spieler zu bestrafen, wird fehlerhaftes Verhalten weniger verlockend, so dass die Spieler kooperativer werden.

- Agenten, die erfolgreich eine Strafe verhängen, können einen persönlichen Vorteil erzielen, indem sie eine Auszahlung erhalten, die von dem bestraften Spieler abgezogen wird, so dass die Agenten einen Anreiz haben den Strafmechanismus zu nutzen und nicht Trittbrettzufahren.

In verschiedenen Experimenten wird gezeigt, dass diese Form von Sanktionierungen kooperative Strategien für Multiagentensysteme mit mehr als 100 lernenden Agenten fördert. Es zeigt sich außerdem, dass die Nutzung des Mechanismus eine dominante Strategie definiert, da ein Agent, der lernt zu bestrafen, eine höhere Belohnung erzielen kann als ein Agent, der den Strafmechanismus nicht nutzt.

5.3.2 Verwandte Arbeiten

Die Schaffung von Anreizsystemen in sozialen Dilemmas zur Steigerung der Kooperationsbereitschaft ist Untersuchungsgegenstand der Spieltheorie. Es lassen sich selektive und sanktionierende Anreizmechanismen unterscheiden [39]. Selektive Anreize beschreiben Ansätze, die versuchen, Kooperation positiv zu fördern, z.B. durch monetäre Belohnungen, um den individuellen Verbrauch von gemeinsam genutzten Ressourcen wie Wasser oder Strom zu reduzieren [51, 94]. Sanktionierende Mechanismen funktionieren im Gegensatz dazu auf der Grundlage von Bestrafungen. Dabei ist das Ziel unkooperatives Verhalten durch Bestrafungen zu reduzieren. Auch Experimente mit Menschen deuten darauf hin, dass Sanktionierungen ein wirksames Mittel sein können, um fehlerhaftes Verhalten zu verringern [15, 40]. Dabei kann das Design des Sanktionierungsmechanismus selbst jedoch eine kritische Rolle spielen. In Experimenten mit Menschen wurden Sanktionierungen erprobt, bei denen die Teilnehmer eine Gebühr zahlen mussten, um andere unkooperative Spieler zu bestrafen [37]. Dadurch ergibt sich u.U. ein soziales Dilemma zweiter Ordnung: Bestrafungen werden vermieden, um individuelle Kosten zu vermeiden (Trittbrettfahrer). Im Bereich des Multiagent Reinforcement Learning gab es dazu andere Ansätze: Perolat et al. [66] haben ein Modell untersucht, in dem mehrere Agenten in einer Grid-Welt leben mit dem Ziel, ihre Belohnung durch das Sammeln einer gemeinsamen Ressource (Äpfel) zu erhöhen. Die Agenten können sich mittels einer speziellen Aktion gegenseitig bestrafen. Der bestrafte Agent wird für 25 aufeinanderfolgende Schritte aus dem Spiel verbannt in denen er also keine Äpfel sammeln kann. Dem bestrafenden Agenten entstehen also keine Kosten, sondern er erhöht indirekt auch noch seine eigene zu erwartende Belohnung, da nun weniger Konkurrenz um die Äpfel besteht.

Der in diesem Abschnitt vorgestellte Sanktionierungsansatz ermöglicht die direkte Beeinflussung der Belohnungen anderer Agenten. Der bestrafende Agent kann durch eine erfolgreiche Bestrafung sogar direkt seine eigene Belohnung erhöhen. Dadurch soll ein soziales Dilemma zweiter Ordnung verhindert werden. Er unterscheidet sich somit auch von dem Ansatz in [66], da dort die Belohnungen nur indirekt beeinflusst werden können.

5.3.3 Ansatz

Der im Folgenden beschriebene Ansatz ist inspiriert von Arbeiten, die darauf hindeuten, dass Sanktionierungsmechanismen dazu beitragen können kooperative Verhaltensweisen zwischen unabhängigen Individuen zu etablieren. Schlüsselfaktoren für entstehende Kooperation zwischen Menschen sind Verpflichtungen, die Möglichkeit zur gegenseitigen Überwachung von Verhaltensweisen sowie die Möglichkeit, diejenigen zu bestrafen, die fehlerhaftes Verhalten an den Tag legen [62]. Hier wird auf diesen Erkenntnissen aufgebaut, indem den Agenten die Möglichkeit gegeben wird sich gegenseitig zu bestrafen, um andere von fehlerhaftem Verhalten abzuschrecken, was wiederum bedeutet, ins-

gesamt die Kooperation zu erhöhen. Die Analogie zu den bisher betrachteten marktbasierten Ansätzen liegt darin, dass eine Transaktion nun keinen positiven Transfer einer Belohnung zwischen zwei Agenten definiert, sondern durch eine Transaktion einem Agenten ein Teil seiner individuellen Belohnung abgezogen werden kann. Durch eine Transaktion verliert also der bestrafte Agent einen Teil seiner Belohnung, wohingegen der Agent, der erfolgreich bestraft, etwas dazu gewinnt.

Formal wird dazu ein N-Spieler Normalform-Spiel $\Gamma = (\mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (r_i)_{i\in\mathcal{N}})$, bestehend aus der Menge Agenten $\mathcal{N}$, einer Menge an Aktionen $\mathcal{A}_i$ für jeden Agenten $i \in \mathcal{N}$, sowie einer Belohnungsfunktion $r_i : \Pi_{i\in N}\mathcal{A}_i \to \mathbb{R}$, für jeden Agenten $i \in \mathcal{N}$ um die folgenden Komponenten erweitert:

- Eine Menge von Bestrafungsaktionen $\mathcal{A}^i_{\text{san}}$ für jeden Agenten $i \in \mathcal{N}$.
- Einem Bestrafungswert $p \in \mathbb{R}_{<0}$

Das in dieser Art erweiterte Normalform-Spiel wird als *Penalty-Game* bezeichnet. Über die Bestrafungsaktionen ist eine Sanktionierung anderer Agenten mit dem Bestrafungswert p möglich. Die Agenten im Penalty-Game wählen also neben ihrer eigentlichen Umweltaktion $e \in \mathcal{A}$ auch eine Bestrafungsaktion x aus der Menge der Bestrafungsaktionen $\mathcal{A}_{\text{san}}$. Die Granularität, mit der Agenten angeben können, wen sie sanktionieren möchten und welche Aktion sie bestrafen wollen, kann dabei über die Menge der Bestrafungsaktionen gesteuert werden. Im allgemeinen Fall kann Agent i jeden anderen Agenten $j \in \mathcal{N}$ für jede beliebige ausgeführte Aktion $u \in \mathcal{A}$ sanktionieren. Im Fall zweier Agenten lässt sich die Aktion des Agenten i somit als Tupel $u_i = (e_i, x_{i,j})$ darstellen, wobei $e_i \in \mathcal{A}^i$ die ausgeführte Umweltaktion von i ist und $x_{i,j} \in \mathcal{A}^i_{\text{san}}$ die zu bestrafende Aktion von Agent j definiert. Der zu sanktionierende Agent j wird gezwungen, die Strafe p zu zahlen, wenn seine Umweltaktion e_j gleich der definierten Bestrafungsaktion $x_{i,j}$ ist, in diesem Fall sind die Belohnungen für i und j:

$$r'_i(..., x_{i,j}, ..., e_j, ...) = r_i + |p * \delta_{x_{i,j},e_j}|$$
$$r'_j(..., x_{i,j}, ..., e_j, ...) = r_j - |p * \delta_{x_{i,j},e_j}|$$

wobei $\delta_{i,j}$ das Kronecker-Delta darstellt, mit $\delta_{k,l} = 1$ wenn $k = l$ und $\delta_{k,l} = 0$ falls $k \neq l$.

5.3.3.0.1 Matching Das Wachstum der Aktionsräume ist durch die oben beschriebenen Erweiterungen exponentiell in der Anzahl der Agenten: $|\mathcal{A}^i| = |\mathcal{A}^i| * \Pi_{j\in\mathcal{N}-1}|\mathcal{A}^j_{\text{san}}|$. Um die Komplexität für Anwendungen mit vielen Agenten zu reduzieren, wird der Sanktionierungsmechanismus mit einem Matching-Verfahren kombiniert, wodurch der Aktionsraum konstant bleibt. Die Idee ist, dass jeder Agent i zu jedem Zeitschritt des Spiels mit einem festen Partner j gematcht wird, so dass nur i und j sich gegenseitig zu diesem Zeitschritt sanktionieren können. Durch das Matching bleibt die Größe des Aktionsraums konstant mit der Anzahl der Agenten, da dadurch gilt $|\mathcal{A}^i| = |\mathcal{A}^i \times \mathcal{A}^i_{\text{san}}|$. Das

Matching selbst kann dabei nach unterschiedlichen Kriterien erfolgen wie beispielsweise fest (der Partner bleibt in jeder Iteration gleich), randomisiert (in jeder Iteration erfolgt eine zufällige Wahl des Partners), regelbasiert (Partner werden nach der Wahl ihrer bisherigen Aktionen gematcht). Für die Evaluation dieses Konzepts wird das Matching randomisiert durchgeführt, so dass in den iterierten Versionen der sozialen Dilemmas, die für die Bewertung verwendet werden, zu jedem Zeitschritt des Spiels zwei Agenten zufällig zusammengebracht werden.

5.3.4 Evaluation

In diesem Abschnitt werden nun Experimente mit zwei Spielern in den sozialen Dilemmas Chicken, Stag Hunt und dem Prisoner's Dilemma beschrieben. Anschließend werden die Ergebnisse in dem N-Spieler Prisoner's Dilemma mit bis zu 128 Agenten beschrieben.

5.3.4.0.1 Agenten und Training Für das Training wird jeder Agent als unabhängige Instanz des tabellarischen Q-Learning-Algorithmus modelliert. Jeder Agent aktualisiert seine Q-Funktion basierend auf individuellen Erfahrungstupeln (s, a, r, s'), bestehend aus Zustand s, Aktion a, Belohnung r, sowie Folgezustand s'. Jedes Spiel wird als iteriertes Normalform-Spiel gespielt, d.h. das Spiel wird sequentiell für eine feste Anzahl an Schritten ausgeführt. Zum Training wurden die sozialen Dilemmas für 4.000 Schritte (genannt Episoden) wiederholt, bevor das Spiel neu beginnt. Es wird eine Lernrate von $\alpha = 0,2$ für die sozialen Dilemmas mit zwei Spielern und eine Lernrate von $\alpha = 0,008$ für das Spiel mit N Spielern verwendet. Die Explorationsrate ϵ wird im Verlauf aller Schritte verringert, beginnend mit $\epsilon_0 = 1,0$ und linear abnehmend bis $\epsilon_{4000} = 0,0001$. In allen Experimenten wird mit einer Diskontierung von $\gamma = 0,9$ gearbeitet.

Chicken	C	D
C	3, 3	1, 4
D	4, 1	0, 0

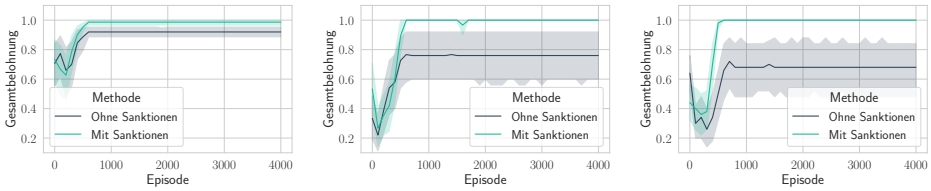
Stag Hunt	C	D
C	4, 4	0, 3
D	3, 0	1, 1

Prisoners	C	D
C	1, 1	-0.5, 1.5
D	1.5, -0.5	0, 0

Abbildung 5.9: Die drei sozialen Dilemmas: Chicken, Stag Hunt und das Prisoner's Dilemma.

5.3.4.1 Soziale Dilemmas mit zwei Spielern

Um die Wirkung des vorgeschlagenen Sanktionierungs-Mechanismus in sozialen Dilemmas mit zwei Spielern zu untersuchen, werden drei der bekanntesten sozialen Dilemmas verwendet: die Spiele *Chicken*, *Stag Hunt* und das *Prisoner's Dilemma*. Die Spiele in Matrix-Form sind in Abbildung 5.9 dargestellt.



(a) Chicken (b) Stag Hunt (c) Prisoner's Dilemma

Abbildung 5.10: Erzielte Gesamtbelohnung in den drei sozialen Dilemmas Chicken, Stag Hunt, und dem Prisoner's Dilemma.

Es ist bekannt, dass kooperative Strategien wie *Tit-for-Tat*, die auf dem Prinzip der Reziprozität beruhen, zwischen eigennützigen Individuen allein durch das Wissen entstehen können, dass man dem Gegenüber auch zukünftig wieder begegnen könnte und sich daher besser kooperativ zeigt, um zukünftigen Vergeltungsschlägen zu entgehen [7]. Das Erlernen solcher Strategien setzt jedoch voraus, dass die Agenten in der Lage sind ein Modell ihres Gegenübers zu erstellen, so dass potentielle Reaktionen des Gegenübers abgeleitet werden können, wie es beispielsweise im *Opponent Modelling* gemacht wird. Werden Strategien mittels modellfreiem Reinforcement Learning in Multiagentensystemen erlernt, ist es jedoch unwahrscheinlich, dass Kooperation basierend auf Reziprozität zustande kommt, da bei rein reaktiven Trainingstechniken andere Agenten gar nicht als eigenständige Entscheidungsträger wahrgenommen werden. Die in dieser Form trainierten Agenten lernen daher nur mit geringer Wahrscheinlichkeit kooperatives Verhalten, insbesondere, wenn geteilte Ressourcen knapp sind [45].

Zunächst wird daher überprüft, welchen Grad an Kooperation das unabhängige Trainieren der Agenten in den drei beschrieben sozialen Dilemmas hervorbringt. Die Ergebnisse sind in Abbildung 5.10 dargestellt. Jedes Szenario wurde für 100 unabhängige Trainingsläufe trainiert, wobei jeder Trainingslauf 4.000 Schritte umfasst. Um die Ergebnisse zwischen den drei sozialen Dilemmas vergleichbar zu machen, wurden die Belohnungen zwischen 0 und 1 normalisiert, so dass 1 der maximal erreichbaren Gesamtbelohnung entspricht.

Es lassen sich Unterschiede bezüglich des erzielten Gesamtergebnisses (der Kooperationsbereitschaft) zwischen den einzelnen Spielen identifizieren. Im Spiel Chicken ist der Unterschied zwischen den beiden Verfahren bezüglich der erlernten Kooperation am geringsten, doch schafft es kein Verfahren, in allen Läufen zu kooperieren, was daran liegen kann, dass beidseitige Kooperation in diesem Spiel kein Nash-Gleichgewicht darstellt. Interessanterweise fällt die durchschnittlich erzielte Kooperationsbereitschaft im Spiel Stag Hunt geringer aus, obwohl beidseitige Kooperation hier ein Nash-Gleichgewicht ist. Die zwei Nash-Gleichgewichte in reinen Strategien und das dadurch erzeugte Koordinationsproblem zwischen den Agenten, scheinen das Erlernen des Optimums bei Stag Hunt zu erschweren. Die insgesamt geringste Kooperationsbereitschaft

ergibt sich im Prisoner's Dilemma, da hier gleich zwei Effekte die Kooperationsbereitschaft negativ beeinflussen: Gier und Angst. Der Umstand, dass das einzige existierende Nash-Gleichgewicht des Spiels auf beidseitigem nicht Kooperieren liegt, erschwert offensichtlich auch die Erlernbarkeit kooperativer Strategien mittels unabhängigem Reinforcement Learning.

Im Vergleich dazu werden die gleichen Trainingsläufe nun unter Verwendung des beschriebenen Sanktionierungsmechanismus durchgeführt. Dazu werden in allen drei sozialen Dilemmas die Aktionsräume der Agenten um die Bestrafungsaktionen erweitert, so dass Agent i Agent j, basierend auf j's gespielter Aktion, bestrafen kann. Der bestehende Aktionsraum $\mathcal{A} = \{C, D\}$ wird somit auf $\{(C,-),(C,C),(C,D),(D,-),(D,C),(D,D)\}$ erweitert, wobei die x-Komponente des Tupels (x, y) die Aktion des Spielers angibt, die der Spieler selbst ausführen möchte und die y-Komponente die Aktion definiert, die im Falle der Ausführung durch den anderen Agenten bestraft werden soll. Für alle drei Spiele wird ein Bestrafungswert von $p = -2$ verwendet. Die Ergebnisse der gleichen Versuchsreihe mit integriertem Sanktionierungsmechanismus sind in Abbildung 5.10 dargestellt. Die Chicken-Resultate werden durch die Erweiterung geringfügig verändert, da hier bereits ein recht hoher Grad an Kooperationsbereitschaft erreicht wurde. Der Unterschied wird deutlicher im Spiel Stag Hunt, wo eine Verbesserung durch die Integration des Sanktionierungsansatzes erzielt wird und die Agenten mit Bestrafungen lernen, immer zu kooperieren. Im Prisoner's Dilemma ist der Unterschied zwischen den Verfahren am deutlichsten: durch die Hinzunahme der Sanktionierungen kann ein optimales Ergebnis erzielt werden.

5.3.4.1.1 Bestrafungswert Die Ergebnisse in den vorangegangenen Experimenten wurden mit einem Bestrafungswert von $p = -2$ erzielt. Dieser Wert wurde durch Testen verschiedener Werte ermittelt. Es stellt sich daher die Frage, inwiefern der Bestrafungswert das Ergebnis beeinflusst bzw. inwiefern andere Werte das Ergebnis verändern. Um diese Frage empirisch zu untersuchen, werden verschiedene Werte im Intervall $[-5, 10]$ für das Prisoner's Dilemma evaluiert. Insgesamt wurden dafür gleichverteilt 150 verschiedene Bestrafungswerte im Intervall $[-5, 10]$ getestet und für jeden Wert die erzielte Gesamtbelohnung in zehn unabhängigen Trainingsläufen ermittelt. Die Ergebnisse (Mittelwert und Konfidenzintervalle zum 95 % Konfidenzniveau) sind in Abbildung 5.11 dargestellt. Alle getesteten Werte mit $p \leq -1,8$ führen nahezu durchgängig zu optimalen Ergebnissen. Für größere Werte fällt das Ergebnis jedoch rapide ab und es zeigt sich von da an ein insgesamt niedrigeres Niveau mit starken lokalen Schwankungen in beide Richtungen.

5.3.4.1.2 Analyse der Dynamiken mittels α-*Rank* Wie oben dargestellt, ermöglicht die Erweiterung des Spiels mittels Sanktionierungen den Agenten, ein kooperatives Gleichgewicht zu erlernen. Es stellt sich die Frage, weshalb kooperative Strategien in dem erweiterten Spiel dominanter werden und so-

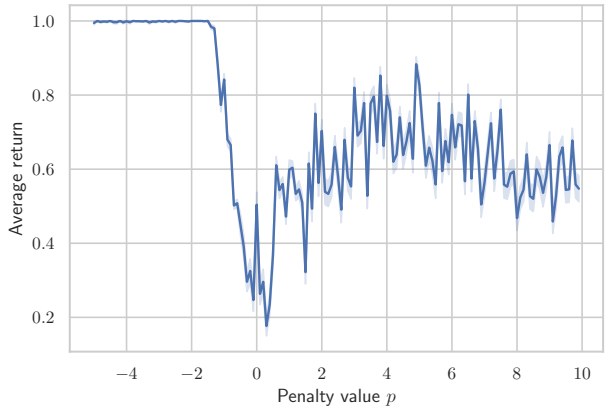


Abbildung 5.11: Die Höhe des Bestrafungswertes p beeinflusst die Höhe der erzielten Gesamtbelohnung wesentlich (hier für das *Prisoner's Dilemma*).

mit leichter gelernt werden können. Zur Analyse der sich ergebenden Lerndynamiken wird das Analysemodell α-*Rank* [60] eingesetzt. α-*Rank* ist ein populationsbasierter Algorithmus zur Analyse der zeitdiskreten Dynamiken, die sich ergeben, wenn jede mögliche Strategie (Aktion) des Spiels als Population modelliert wird, die sich, basierend auf deren relativer Fitness, gegenüber anderen Strategien entwickelt. Das Ergebnis der Analyse ist ein Graph, der die Entwicklung der Populationen beschreibt. Diese Analyse wird für das Prisoner's Dilemma mit und ohne Sanktionierungsmechanismus durchgeführt. Der zugehörige Graph für das Prisoner's Dilemma ohne Sanktionierungen ist in Abbildung 5.12 dargestellt.

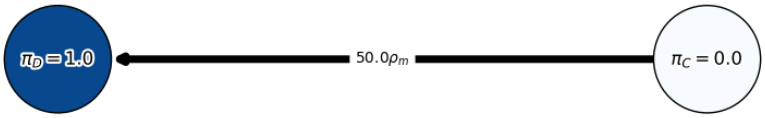


Abbildung 5.12: α-*Rank*-Graph des Prisoner's Dilemma ohne Sanktionen.

Da das Spiel zwei Aktionen hat, hat der Graph zwei Knoten und eine verbindende Kante. Die Beschriftung der Knoten gibt die Strategie an (π_D steht dabei für die nicht kooperative Aktion D und π_C für Kooperation C). Die Zeit, die die Populationen in jeder Strategie durchschnittlich verbringen, wird als die Masse der stationären Verteilung in diesem Knoten quantifiziert und ist als numerischer Wert hinter der Knotenbeschriftung angegeben. Die Kanten zwischen den Knoten entsprechen den Flusswahrscheinlichkeiten für Paare von Strategien, so dass die Kantenrichtung den Fluss der Individuen von einer Strategie zu einer geeigneteren Strategie anzeigt. Für den Fall des Prisoner's

Dilemma zeigt die Kante von π_C nach π_D, die Individuen der Population π_C sind denen aus Strategie π_C also unterlegen. Auch ist alle Masse der stationären Verteilung in der nicht kooperativen Strategie D akkumuliert, so dass nur Population π_D überleben kann. Dieses Ergebnis deckt sich mit dem einzigen Nash-Gleichgewicht im Prisoner's Dilemma, das ebenfalls in der Strategie (D, D) liegt.

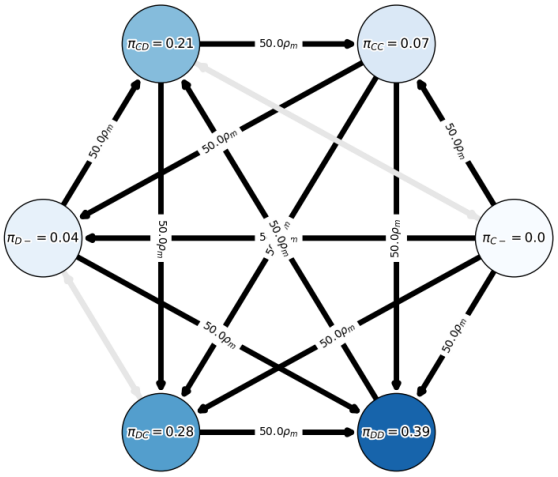


Abbildung 5.13: Graph zur α-*Rank*-Analyse des Prisoner's Dilemma mit Sanktionierungen.

Abbildung 5.13 zeigt demgegenüber den Graphen für das Prisoner's Dilemma mit Sanktionierungsmechanismus. Der Graph enthält nun entsprechend sechs Knoten, da es sechs Aktionen gibt. Hier teilt sich die Masse der stationären Verteilung (das heißt die Wahrscheinlichkeit, mit der die Populationen in einer bestimmten Strategie verweilen) nun auf mehrere Knoten auf, wobei auch ein Teil der Wahrscheinlichkeitsmasse auf kooperative Strategien fällt. Jeder Knoten hat eine wegführende Kante, so dass also keine Strategie überlegen zu allen anderen Strategien ist. Für die Lernbarkeit mittels Reinforcement Learning scheint es jedoch ausreichend zu sein, dass es kein eindeutiges suboptimales Gleichgewicht mehr gibt, so wie im Prisoner's Dilemma ohne Bestrafungen. Obwohl im Fall mit Bestrafungen keine Strategie alle anderen dominiert, konvergiert der Lernprozess der Agenten stabil zu einem optimalen kooperativen Ergebnis. Die Tatsache, dass der Lernprozess trotz Zyklen im Graph konvergiert, könnte auf die abnehmende Explorationsrate der Agenten zurückzuführen zu sein, weil dadurch im Zeitverlauf weniger Aktionen zufällig erprobt werden.

5.3.4.2 Soziales Dilemma mit N-Spielern

Die Evaluation wird nun erweitert auf ein soziales Dilemma, das mit beliebiger Anzahl von Agenten durchgeführt werden kann. Dazu wird eine N-Spieler-Variante des *Public Good Games* verwendet, sowie in [9] beschrieben. Das Spiel

umfasst eine Menge Agenten $\mathcal{N}$, die jeweils zwischen zwei möglichen Aktionen C und D wählen können. Aktion C bedeutet ein Agent kooperiert, wodurch er einen positiven Beitrag P zu der Bereitstellung eines öffentlichen Gutes beiträgt, während ein Agent durch die Wahl von Aktion D keinen Beitrag leistet. Das öffentliche Gut, das durch alle kooperierenden (also beitragenden) Agenten bereitgestellt wird, wird gleichmäßig auf alle Agenten (auch nicht kooperierende) verteilt, d.h. die Kosten zur Bereitstellung des öffentlichen Gutes werden nur von den kooperativ handelnden Agenten getragen, wohingegen die nicht kooperierenden Agenten keine Kosten übernehmen (Trittbrettfahrer). Die Belohnungsfunktionen der kooperierenden Agenten (r_c) bzw. der nicht kooperierenden Agenten (r_d) sind:

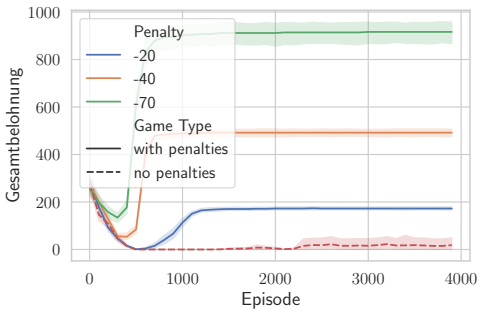
$$r_c = \frac{f * c * P}{|\mathcal{N}|} - P, r_d = \frac{f * c * P}{|\mathcal{N}|}$$

wobei f ein frei wählbarer Skalierungsfaktor ist, P ist der Anteil, den jeder Kooperateur zum öffentlichen Gut beiträgt und c ist die Anzahl der Kooperateure. Für die hier vorgestellten Experimente wurde folgende Parameter verwendet: Der Beitrag P, den jeder kooperative Agent einbringt ist $P = 1$. Der Skalierungsfaktor wurde auf $f = 2$ gesetzt, so wie in [9].

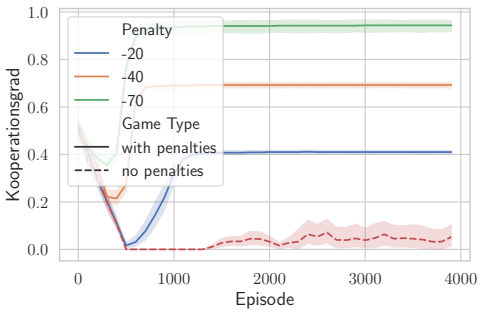
5.3.4.2.1 Ergebnisse Die Ergebnisse umfassen Trainingsläufe von Agenten mit und ohne Sanktionierungsmechanismus. Insgesamt wurde das Szenario mit 32, 64 und 128 Agenten getestet, wobei jeder Agent eine eigene Instanz des tabularen *Q-Learning*-Algorithmus darstellt. Sämtliche Parameter für das Q-Learning wurden dabei für die sozialen Dilemmas mit zwei Spielern belassen, lediglich die Lernrate wurde auf $\alpha = 0,008$ reduziert. Zunächst wird die erzielte Gesamtbelohnung dargestellt (Summe aller Belohnungen von kooperativen und nicht kooperativen Agenten), sowie der Grad an erreichter Kooperation (Anteil Kooperateure an Gesamtanzahl Agenten). Abbildung 5.14 zeigt die erzielte Gesamtbelohnung und die Kooperationsrate über den Verlauf von 4.000 Trainingsschritten. Es werden unterschiedliche Bestrafungswerte p getestet, wobei für jeden Bestrafungswert 100 unabhängige Trainingsläufe durchgeführt werden, so dass die Abbildungen Mittelwerte und Konfidenzintervalle zum Konfidenzniveau von 95% darstellen. Die Werte für p liegen dabei in dem Intervall $[-20, 300]$ und sind entsprechend der Gesamtanzahl der Agenten skaliert.

In allen Szenarien ist die Gesamtbelohnung und der Grad an Kooperation signifikant höher unter Verwendung des Sanktionierungsverfahrens. Die Höhe der erzielten Steigerung hängt von der Höhe des gewählten Strafwerts p ab, wobei höhere Strafen (niedrigere Werte für p) mit höheren Gesamtbelohnungen und einer höheren Kooperationsrate einhergehen. Wenn Sanktionierungen möglich sind, erreicht die Kooperationsrate in allen Experimenten Werte über 90 Prozent, während ohne Strafen stabile Kooperation scheitert, wie die niedrigen Belohnungen und die Rate der Kooperierenden zeigen.

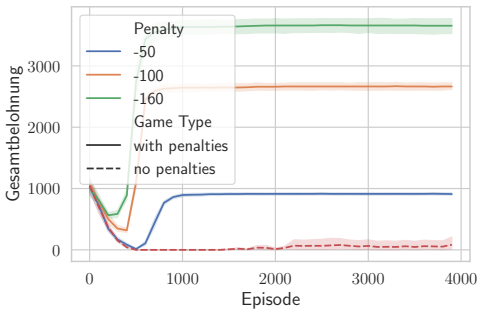
Bisher wurde die erzielte Gesamtbelohnung und der Kooperationsgrad be-



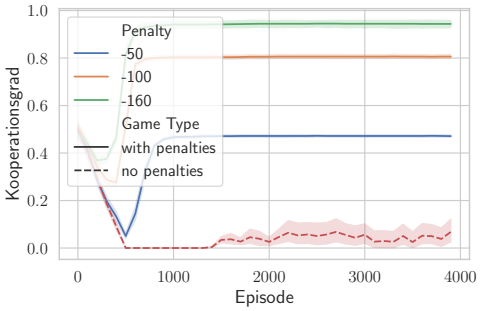
(a) Belohnung (32 Agenten)



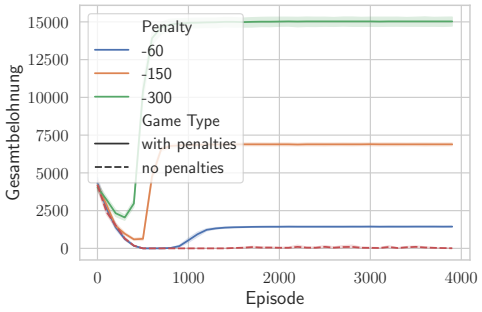
(b) Kooperation (32 Agenten)



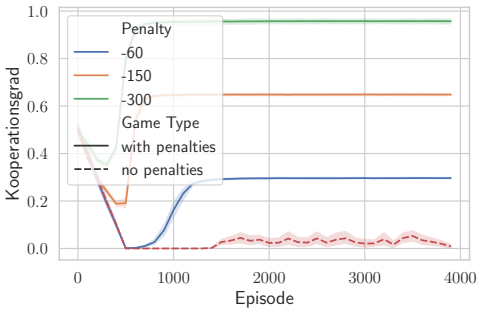
(c) Belohnung (64 Agenten)



(d) Kooperation (64 Agenten)

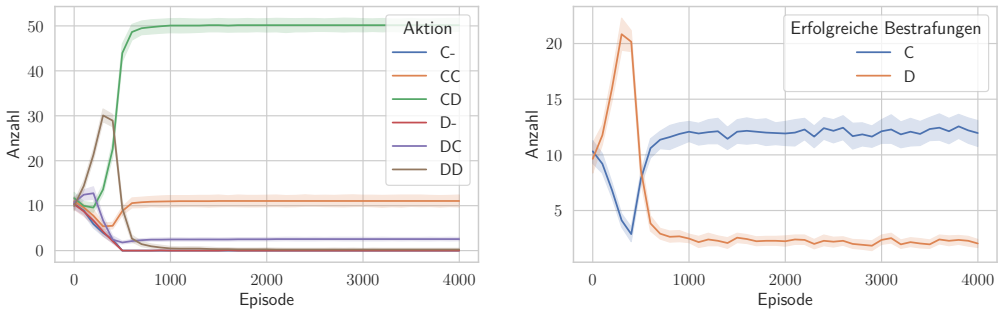


(e) Belohnung (128 Agenten)



(f) Kooperation (128 Agenten)

Abbildung 5.14: Gesamtbelohnung (Summe der Belohnungen der Agenten) und Kooperationsrate (Anteil der Kooperateure bei jedem Schritt) für 32, 64 und 128 Agenten im N-Spieler Public Good Game. Dargestellt sind Mittelwerte und die 95% Konfidenzintervalle.



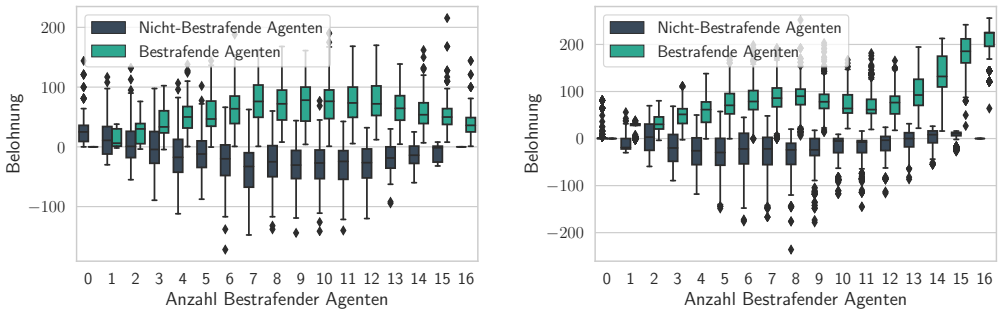
(a) Gespielte Aktion

(b) Erfolgreich durchgeführte Bestrafungen

Abbildung 5.15: Gespielte Aktionen und Anzahl der erfolgreichen Sanktionierungen für 64 Agenten im N-Spieler *Public Good Game* (Mittelwert und die 95% Konfidenzintervalle).

richtet. Einen genauen Überblick, welche Aktionen zu welchem Zeitpunkt im Training der Agenten gespielt werden und wie viele Sanktionen tatsächlich verhängt wurden, gibt Abbildung 5.15. Das Szenario umfasst 64 Agenten, und für die Ergebnisse wurden 25 unabhängige Durchläufe gemittelt. Die Ergebnisse zeigen, dass in der frühen Phase des Lernens (d.h. für Episode < 500) ein starker Anstieg der Aktion DD stattfindet, was bedeutet, dass es in dieser Phase attraktiv ist, selbst nicht zu kooperieren und gleichzeitig andere für nicht kooperatives Verhalten zu bestrafen. Nach dieser Phase wird die Aktion CD (kooperativ-sein und nicht kooperatives Verhalten bestrafen) immer attraktiver und wird für den Rest des Trainings von etwa 50 der insgesamt 64 Agenten als dominante Strategie gewählt. Es verbleibt eine kleinere Gruppe von Agenten, die die Strategie lernen selbst kooperativ zu sein und andere für Kooperation zu bestrafen (etwa 10 Agenten) und eine weitere kleine Gruppe, die lernt nicht zu kooperieren und andere für Kooperation zu bestrafen (durchschnittlich 4 Agenten gegen Ende des Trainings). Diese Strategien spiegeln sich auch in der Anzahl der erfolgreich durchgeführten Bestrafungen wider, wie in Abbildung 5.15b dargestellt. Zu Beginn gibt es einen starken Anstieg erfolgreicher Bestrafungen nicht kooperativer Agenten, die dann aber schnell weniger werden, da nur noch sehr wenige nicht kooperative Agenten übrig sind, so dass die Bestrafung dieser Gruppe unattraktiv wird. Die kleine Gruppe von Agenten, die auf die Bestrafung von Kooperateuren spezialisiert ist, führt zu einem konstanten Anteil an erfolgreichen Strafen für Kooperation nach etwa Episode 500.

5.3.4.2.2 Robustheit Zur Untersuchung der Robustheit der Sanktionierungen werden *Schelling*-Diagramme verwendet. Dabei geht es darum darzustellen, wie sich die Belohnungen der Agenten verändern, wenn sich einzelne Agenten entscheiden den Sanktionierungsmechanismus zu nutzen. Dazu werden



(a) Frühe Trainingsphase (b) Späte Trainingsphase

Abbildung 5.16: Shelling-Diagramme des Public Good Games für 16 Agenten. Dargestellt sind die Belohnungen für eine unterschiedliche Anzahl bestrafender Agenten. Links: frühes Training (Episode $<$ 500), rechts: spätes Training (Episode $>$ 3000).

Trainingsläufe mit einer unterschiedlichen Anzahl von Agenten, die sanktionieren können, durchgeführt. Die Menge der Agenten die sanktionieren, sei mit S gekennzeichnet, wobei es insgesamt 16 Agenten gibt. Es werden dann die aggregierten Belohnungen aller sanktionierenden Agenten mit den Belohnungen der nicht-sanktionierenden Agenten verglichen, sowie in Abbildung 5.16 dargestellt. Ganz links im Diagramm gibt es keine sanktionierenden Agenten, das entspricht also dem Spiel ohne Sanktionierungsmechanismus. Am äußersten rechten Punkt enthält das Spiel nur sanktionierende Agenten ($|S| = 16$).

Um nun auch zu untersuchen, wie sich die relativen Belohnungen von sanktionierenden Agenten und nicht-sanktionierenden Agenten über den Verlauf des Trainings verändern, wird jeweils ein Schelling-Diagramm aus einer frühen Trainingsphase (Episode $<$ 500) mit einem Schelling-Diagramm aus einer späten Trainingsphase (Episode $>$ 3.000) verglichen. Es zeigt sich, dass während des späten Trainings die Belohnungen von Agenten, die andere bestrafen können, für alle Werte von $|S|$ höher sind als die der Agenten, die nicht sanktionieren können. Es ist also zu jedem Zeitpunkt individuell rational, den Sanktionierungsmechanismus zu nutzen, wenn er verfügbar ist. Wenn mehr als die Hälfte aller Agenten bestrafen können ($|S| > 8$), sinkt die Belohnung lokal für alle sanktionierenden Agenten leicht, bis sie sich bei $|S| > 11$ wieder erholt. Dieser Effekt scheint jedoch lokal begrenzt zu sein, da die Belohnungen aller sanktionierenden Agenten ab $|S| > 11$ wieder zu steigen beginnt und insgesamt alle sanktionierenden Agenten die höchste Belohnung erzielen, wenn alle Agenten sanktionieren können.

5.4 Zusammenfassung

In diesem Kapitel wurden zwei weitere relevante Themen aufgegriffen, die für die Kooperation zwischen eigennützigen Agenten relevant sind. Im ersten Abschnitt wurden Verträge vorgestellt, die zwischen jeweils zwei Agenten abgeschlossen werden können. Verträge regeln sowohl das Verhalten, das die Vertragspartner während der Laufzeit eines Vertrages einhalten müssen als auch die Belohnungen, die den Teilnehmern dadurch zukommen. Die Auszahlungen werden dabei über die Q-Wertefunktionen ermittelt. Ein Vertrag stimmt einen Agenten also mindestens indifferent zwischen seiner eigenen besten Policy und der Policy des Vertrags.

Im zweiten Teil dieses Kapitels wurde eine weitere Klasse an Szenarien betrachtet, die eine besondere Herausforderung an kooperatives Verhalten stellt. Diese sogenannten sozialen Dilemmas erschweren ein direktes Anwenden der zuvor behandelten Mechanismen, die auf einer positiven Incentivierung der Akteure beruhen. Inspiriert von Erkenntnissen aus verhaltenswissenschaftlichen Experimenten, wurde das konditionale Action Trading, durch die Verwendung eines negativen Preises, als Sanktionierungsmechanismus interpretiert. In den drei kanonischen sozialen Dilemmas Stag Hunt, Chicken und dem Prisoner's Dilemma sowie in dem N-Spieler Public Good Game, konnte gezeigt werden, dass Peer-to-Peer basierte Bestrafungen den Grad an Kooperation erheblich erhöhen können.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurden Methoden zur Unterstützung kooperativer Verhaltensweisen in lernenden Multiagentensystemen vorgestellt. Die Methoden basieren auf bilateralen Austauschmechanismen zwischen den Agenten. Grundlage aller Ansätze ist das stochastische Marktspiel (siehe Kapitel 3), welches das formale Konzept vorgibt. Das Konzept erweitert das stochastische Spiel um eine Menge an Marktaktionen sowie eine Marktfunktion. Die Marktaktionen geben den Agenten die Möglichkeit als Marktteilnehmer aktiv zu werden und in Austausch mit anderen Agenten zu treten. Die Marktfunktion definiert auf welcher Grundlage Transaktionen stattfinden und welche Belohnungen aus einem Handel resultieren. Bei allen Ansätzen wurde das emergente Systemverhalten betrachtet: dazu wurden Methoden des Reinforcement Learning eingesetzt und die Entwicklungen mit und ohne Marktmechanismus miteinander verglichen. Da der Preis nicht Teil dieses emergenten Lernprozesses ist, sondern als fester Parameter vorgegeben wird, wurden auch Verhandlungsprozesse zwischen unabhängigen Agenten modelliert und potenzielle Einflüsse auf dieses Ergebnis untersucht. Hierbei zeigt sich, dass die Ergebnisse, die mittels Reinforcement Learning erzielt wurden, sich mit Ergebnissen der Spieltheorie decken, was insofern bemerkenswert ist, als dass in der Spieltheorie andere Annahmen über die Agenten getroffen werden. Hier ist insbesondere die Annahme der Rationalität der Agenten hervorzuheben, die in der Spieltheorie angenommen wird. Es wird dabei davon ausgegangen, dass Agenten sich stets strategisch verhalten und alle verfügbaren Informationen perfekt verarbeiten können. Bei Lernmethoden, die durch Versuch und Irrtum gesteuert werden, müssen diese Annahmen zwangsläufig fallen gelassen werden. Das beide Verfahren auf ähnliche Ergebnisse kommen ist daher in zweifacher Weise interessant: Einerseits zeigt es, dass das Training mittels Reinforcement Learning im Ergebnis trotz irrationaler Aktionswahl während des Trainings im Ergebnis dem rationalen Verhalten spieltheoretischer Agenten gleicht. Was den Einsatz von Reinforcement Learning in Multiagentensystemen weiter plausibilisiert. Anderseits zeigen die durch Reinforcement Learning erzielten Ergebnisse, dass die starken Annahmen der Spieltheorie auch empirisch erhärtet werden können.

In Kapitel 4 werden zwei unterschiedliche Marktformen genauer beleuchtet. Die erste Marktform funktioniert auf Basis von konditionalen Transaktionen zwischen den Agenten, d.h. jeder Handel beinhaltet stets eine Leistung und eine Gegenleistung. Der Ansatz eignet sich daher, um bestimmte Verhaltensweisen

zu incentivieren, da ein Handel auch auf bestimmte Aktionen des Partners konditioniert werden kann. Die zweite Klasse an Marktformen erfordert keine Konditionierung, so dass keine direkte Gegenleistung für einen Handel erforderlich ist. Dieses Prinzip ermöglicht es den Akteuren ihre Belohnungsfunktionen aneinander anzugleichen, da ein Agent Anteile seiner eigenen erwarteten Belohnung an andere Agenten ausgeben kann. Die Anteilseigner sind nun ihrerseits durch ihren Anteil daran interessiert, dass der Herausgeber der Anteile sein Ziel erreicht, so dass auch sie von den entstehenden Belohnungen profitieren. Die Evaluation der Ansätze in verschiedenen Umgebungen, zeigt, dass die Marktmechanismen den erreichten Grad an Effizienz des Systems deutlich erhöhen können.

Kapitel 5 stellt einen weiteren marktbasierten Ansatz vor, der explizite Vertragsabschlüsse zwischen den Agenten ermöglicht. Ein Vertrag definiert dabei sowohl das Verhalten der Vertragsteilnehmer als auch die daraus resultierenden Belohnungen. Bei diesem Ansatz werden die Belohnungen aus zuvor gelernten Wertefunktionen der Agenten ermittelt, so dass eine Kompensation gemäß der entgangenen Belohnungen, durch die Wahl einer individuell optimalen Aktion, gewählt wird. Im zweiten Teil dieses Kapitels wird die Anwendbarkeit der Ansätze auf die Klasse sozialer Dilemmas untersucht. Dabei stellt sich heraus, dass der konditionale Marktmechanismus durch die Wahl eines negativen Preises in einen Bestrafungsmechanismus umgewandelt werden kann. Hier erfolgt die Incentivierung nicht mehr durch den Transfer einer positiven Belohnung, sondern durch den potenziellen Verlust von Belohnungen. Dieser Mechanismus wurde in einem sozialen Dilemma mit bis zu 128 Agenten evaluiert, wobei sich zeigt, dass der erreichte Kooperationsgrad deutlich gesteigert werden kann.

Für die Zukunft ist eine naheliegende Frage, inwiefern Marktprotokolle in realen Anwendungen eingesetzt werden können, um heterogene Teilnehmer durch dezentralen Austausch zu Kooperation zu bewegen. Ein entscheidender Schritt dafür ist die Ermittlung von effizienten Preisen, d.h. Preisen, die von beiden Seiten für eine Transaktion akzeptiert werden. In den vorgestellten Ansätzen wurden Preise manuell festgesetzt, so dass eine Transaktion möglich ist. Es stellt sich jedoch die Frage, ob Preise durch Verhandlungen der Teilnehmer autonom gefunden können. Es ist denkbar, dass Preise gefunden werden können, die die Anzahl der Transaktionen noch erhöhen oder ein verbessertes Training ermöglichen. Vorstellbar wäre hier eine weitere autonome Einheit in den Trainingsprozess zu integrieren, deren Ziel es ist einen optimalen Preis zu finden, so dass möglichst viele Transaktionen zwischen den Agenten realisiert werden. Die Preisfindung unabhängig von den Agenten zu machen, hält die Komplexität der Agenten-Agenten Interaktion gering, da ein Agent so lediglich entscheiden muss, ob für einen gegeben Preis gehandelt werden soll oder nicht.

Ein weiterer Aspekt, der insbesondere vor dem Hintergrund wachsender Systeme helfen kann die Komplexität zu verringern, ist es die Agenten-Interaktionen anhand bestimmter Kriterien weiter zu strukturieren. Ein na-

heliegender Ansatz wäre eine nähebasierte Interaktion, denn es ist realistisch, davon auszugehen, dass nur Agenten in einer bestimmten Nachbarschaft zueinander überhaupt Interesse daran haben mit anderen benachbarten Agenten zu handeln, da sie z.B. um eine nahegelegene Ressource konkurrieren. Dabei könnte eine Ressource selbst Preise definieren, zu denen sie gerade genutzt werden kann und lediglich Agenten in der Nähe der Ressource können Angebote dafür abgeben.

Andere Fragen, die im Kontext von handelnden Agenten relevant werden, betreffen die Ansammlung von Reichtum durch die Agenten und inwiefern unterschiedliche Budgets der Agenten möglicherweise zu effizienten aber unfairen Allokationen führen können. Das ist insbesondere wichtig, wenn Budgets nicht an Trainingsepisoden gebunden sind, sondern kontinuierlich akkumuliert werden können. Durch zu starke Ungleichgewichte könnten Fairness-Bedingungen (jeder Agent hat eine Chance seine Aufgaben zu erfüllen) verletzt werden. Es wäre daher denkbar, dass auch KI-getriebene Märkte, wie reale Märkte, ein gewisses Maß an Regulierung benötigen, da die reine Funktionsweise des Marktes zu einem sozial unerwünschten Ergebnis führt, insbesondere über längere Zeitskalen betrachtet.

Abkürzungsverzeichnis

Abbildungsverzeichnis

2.1 Zwei Spieler Normalform-Spiel mit zwei Aktionen in Matrixform. 9
2.2 Das Prisoner's Dilemma 10
2.3 Verhandlungssituation zwischen zwei Spielern. Der Kreis markiert die Möglichkeiten zur Zusammenarbeit, doch es gibt Interessenskonflikte. 13
2.4 Beim Reinforcement Learning interagiert der Agent wiederholt mit einer unbekannten Umgebung. 15
2.5 Schematischer Aufbau eines *feedforward*-neuronalen Netzes mit zwei versteckten Schichten, drei Eingangssignalen und vier Ausgangssignalen. 20

3.1 Im stochastischen Marktspiel wählen die Agenten sowohl Aktionen aus dem originalen Aktionsraum als auch Marktaktionen, um mit anderen Agenten zu Handeln. 38
3.2 Matrix-Spiel mit Konflikt- und Koordinations-Problem zwischen den Agenten. 42
3.3 Das Konfliktspiel: Der Konflikt zwischen den Spielern lässt sich über den Parameter α steuern. 43
3.4 Vergleich der Ergebnisse im Konfliktspiel mit und ohne Markt. . 45
3.5 Die Verhandlungsdomäne *Divide-The-Grid* modelliert die Aufteilung der Zellen einer Grid-Welt zwischen zwei Agenten. . . . 51
3.6 Käufer-Verkäufer-Verhandlungs-Domäne: Zwei Agenten müssen einen Preis festlegen, um ein Handelsgut zu transferieren. 52
3.7 Detailansicht des Verkäufer-Agenten in der Verkäufer-Käufer-Domäne. 52
3.8 Verteilungen über die zu erwartenden zukünftigen Belohnungen, für die vier möglichen Aktionen des blauen Agenten, in dem dargestellten Zustand. 54
3.9 Verteilungen über die zu erwartenden zukünftigen Belohnungen, für die vier möglichen Aktionen des blauen Agenten, inklusive *CVaR*-Werte für alle Verteilungen. 56
3.10 Das Verhandlungsergebnis wird durch die Risikoaversion der Agenten beeinflusst. Abgebildet sind die Verhandlungsanteile für Agent 1 (links) und Agent 2 (rechts), für unterschiedliche Niveaus der Risikoaversion, modelliert durch unterschiedliche Werte von α. 58

3.11 Verhandlungserfolg (Wahrscheinlichkeit) in der Verkäufer-Käufer-Domäne in Abhängigkeit des Auftretens des Produktes in niedriger Qualität p_l und der Bewertung des Gutes durch den Käufer (U_b). 60

4.1 Das Coin Game: Zwei Agenten konkurrieren um eine Münze. Während rein eigennütziges Verhalten ohne Handel die Agenten zu gierigem Handeln anregt, kann die Einführung eines Marktes dazu beitragen, den erwarteten Wert beider Agenten zu erhöhen. 64
4.2 Der Ansatz Action Trading lässt Agenten Belohnungen gegen bestimmte Aktionen tauschen. Agenten wählen also zusätzlich zu ihren ursprünglichen Aktionen auch Marktaktionen. Ein Handel kommt zustande, wenn ein Angebot für eine Aktion mit einer tatsächlich ausgeführten Aktion übereinstimmt. 67
4.3 Spiel mit 2 Nash-Gleichgewichten. 69
4.4 Das erweiterte Spiel mit Action Trading. 69
4.5 Gesamtbelohnung und Anzahl der Transaktionen des Matrix-Spiels aus Abbildung 4.3. 70
4.6 Aktionswahl mit und ohne Action Trading in einem zwei-Spieler-Spiel. 71
4.7 Das Coin Game: zwei Agenten konkurrieren um Münzen. Sammelt ein Agent die Münze des anderen, erhält dieser eine negative Belohnung. 72
4.8 Durch Action Trading lässt sich das Dilemma im Coin Game auflösen. 72
4.9 Vergleich der Belohnungen im Coin Game mit und ohne Action Trading. 76
4.10 Anteil der korrekt gesammelten Münzen im Coin Game und Gesamtzahl gesammelter Münzen. 76
4.11 Entwicklung der Transaktionen pro Episode im Coin Game. . . 77
4.12 Ergebnisse im Coin Game mit vier Agenten. 78
4.13 Durch das *Shareholding* wird es den Agenten ermöglicht Anteile der eigenen zukünftigen Belohnung an andere auszugeben und Anteile anderer Agenten zu erwerben. 83
4.14 Die Smartfactory: die Agenten (blaue Dreiecke) müssen verschiedene Maschinen abfahren (rote bzw. schwarze Hexagone), um ihre individuellen Aufträge abzuarbeiten. 86
4.15 Erzielte Gesamtbelohnung in der Smartfactory für 4, 8 und 16 Agenten. 88
4.16 Erzielte Gesamtbelohnung in der Raffinerie-Umgebung für 4, 8 und 16 Agenten. 89
4.17 Darstellung der Aufteilung der erzielten Gesamtbelohnung auf Agenten mit niedriger Priorität und hoher Priorität, für die Smartfactory. 90

5.1 DEAL-Ablauf: Zwei Agenten (2 und 3) schließen bilaterale Verträge mit Agent 1, der die Agenten 2 und 3, für deren entgangene erwartete Belohnungen, entschädigen muss. 101
5.2 Vergleich der Gesamtbelohnung in der Smartfactory mit und ohne DEAL, für zwei Agenten und in Abhängigkeit der Ressourcenverfügbarkeit (Anzahl der Zeitschritte, nach der die Maschine nicht nutzbar ist). 104
5.3 Erzielte Belohnung und Anzahl abgeschlossener Verträge, in Abhängigkeit zu den Kosten für Aufträge mit hoher und niedriger Priorität. 105
5.4 Empirisch ermittelte Auszahlungsmatrizen für zwei Agenten mit und ohne DEAL. 106
5.5 Ergebnisse unter Verwendung unterschiedlicher DEAL-Policies. . 108
5.6 t-SNE-Einbettung der letzten verborgenen Schicht eines trainierten Agenten mit Clustern, die nach bestimmten Spielzuständen gefärbt sind, d. h. (a) nach den zugehörigen Prioritäten und (b) nach den erfolgreichen Verträgen, die von diesem Agenten, während einer einzelnen Episode, angeboten wurden. 109
5.7 Erzielte Belohnung und Verteilung der Belohnung für vier, acht und 16 Agenten in der Smartfactory. 112
5.8 Entwicklung der individuellen Belohnungen nach Gesamtanzahl von DEAL-Agenten im System. 114
5.9 Die drei sozialen Dilemmas: Chicken, Stag Hunt und das Prisoner's Dilemma. 118
5.10 Erzielte Gesamtbelohnung in den drei sozialen Dilemmas Chicken, Stag Hunt, und dem Prisoner's Dilemma. 119
5.11 Die Höhe des Bestrafungswertes p beeinflusst die Höhe der erzielten Gesamtbelohnung wesentlich (hier für das *Prisoner's Dilemma*). 121
5.12 α-*Rank*-Graph des Prisoner's Dilemma ohne Sanktionen. 121
5.13 Graph zur α-*Rank*-Analyse des Prisoner's Dilemma mit Sanktionierungen. 122
5.14 Gesamtbelohnung (Summe der Belohnungen der Agenten) und Kooperationsrate (Anteil der Kooperateure bei jedem Schritt) für 32, 64 und 128 Agenten im N-Spieler Public Good Game. Dargestellt sind Mittelwerte und die 95% Konfidenzintervalle. . 124
5.15 Gespielte Aktionen und Anzahl der erfolgreichen Sanktionierungen für 64 Agenten im N-Spieler *Public Good Game* (Mittelwert und die 95% Konfidenzintervalle). 125
5.16 Shelling-Diagramme des Public Good Games für 16 Agenten. Dargestellt sind die Belohnungen für eine unterschiedliche Anzahl bestrafender Agenten. Links: frühes Training (Episode $<$ 500), rechts: spätes Training (Episode $>$ 3000). 126

Literatur

[1] A. K. Agogino und K. Tumer. “Unifying temporal and structural credit assignment problems”. In: *AAMAS*. Bd. 4. 2004, S. 980–987.

[2] G. A. Akerlof. “The market for "lemons": Quality uncertainty and the market mechanism”. In: *The quarterly journal of economics* (1970), S. 488–500. DOI: 10.1017/CBO9780511609381.002.

[3] G. A. Akerlof. “The market for “lemons”: Quality uncertainty and the market mechanism”. In: *Uncertainty in Economics*. Elsevier, 1978, S. 235–251.

[4] N. Anastassacos, S. Hailes und M. Musolesi. “Partner selection for the emergence of cooperation in multi-agent systems using reinforcement learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Bd. 34. 05. 2020, S. 7047–7054. DOI: 10.1609/aaai.v34i05.6190.

[5] N. Anbarci. “Divide-the-dollar game revisited”. In: *Theory and Decision* 50.4 (2001), S. 295–303.

[6] W. B. Arthur. “Complexity economics”. In: *Complexity and the Economy* (2013).

[7] R. Axelrod und W. D. Hamilton. “The evolution of cooperation”. In: *science* 211.4489 (1981), S. 1390–1396. DOI: 10.1126/science.7466396.

[8] T. Baarslag, M. J. Hendrikx, K. V. Hindriks und C. M. Jonker. “Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques”. In: *Autonomous Agents and Multi-Agent Systems* 30.5 (2016), S. 849–898. DOI: 10.1007/s10458-015-9309-1.

[9] J. V. Barbosa, A. H. R. Costa, F. S. Melo, J. S. Sichman und F. C. Santos. “Emergence of Cooperation in N-Person Dilemmas through Actor-Critic Reinforcement Learning”. In: (2020).

[10] E. B. Baum. “Toward a model of intelligence as an economy of agents”. In: *Machine Learning* 35.2 (1999), S. 155–185.

[11] A. L. Bazzan. “Aligning individual and collective welfare in complex socio-technical systems by combining metaheuristics and reinforcement learning”. In: *Engineering Applications of Artificial Intelligence* 79 (2019), S. 23–33. DOI: 10.1016/j.engappai.2018.12.003.

[12] M. G. Bellemare, W. Dabney und R. Munos. “A distributional perspective on reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2017, S. 449–458.

[13] S. J. Brams und A. D. Taylor. “Divide the dollar: Three solutions and extensions”. In: *Theory and Decision* 37.2 (1994), S. 211–231. DOI: 10.1007/BF01079266.

[14] L. Buşoniu, R. Babuška und B. De Schutter. “Multi-Agent Reinforcement Learning: An Overview”. In: *Innovations in multi-agent systems and applications-1*. Springer, 2010, S. 183–221.

[15] M. D. Caldwell. “Communication and sex effects in a five-person Prisoner’s Dilemma Game.” In: *Journal of Personality and Social Psychology* 33.3 (1976), S. 273. DOI: 10.1037/0022-3514.33.3.273.

[16] G. Chalkiadakis und C. Boutilier. “Bayesian reinforcement learning for coalition formation under uncertainty”. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*. IEEE Computer Society. 2004, S. 1090–1097.

[17] G. Chalkiadakis, E. Elkind und M. Wooldridge. “Computational aspects of cooperative game theory”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5.6 (2011), S. 1–168. DOI: 10.2200/S00355ED1V01Y201107AIM016.

[18] M. Chang, S. Kaushik, S. M. Weinberg, T. Griffiths und S. Levine. “Decentralized reinforcement learning: Global decision-making via local economic transactions”. In: *International Conference on Machine Learning*. PMLR. 2020, S. 1437–1447.

[19] R. Chaudhuri, K. Mukherjee, R. Narayanam, R. D. Vallam, A. Kumar, A. Mathur, S. Garg, S. Singh und G. Parija. “Collaborative Reinforcement Learning Model for Sustainability of Cooperation in Sequential Social Dilemmas”. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 2019, S. 1877–1879.

[20] D.-A. Clevert, T. Unterthiner und S. Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[21] A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson und T. Graepel. “Open Problems in Cooperative AI”. In: *arXiv preprint arXiv:2012.08630* (2020).

[22] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris und B. Coppin. “Deep reinforcement learning in large discrete action spaces”. In: *arXiv preprint arXiv:1512.07679* (2015).

[23] T. Ellingsen. "The evolution of bargaining behavior". In: *The Quarterly Journal of Economics* 112.2 (1997), S. 581–602. DOI: 10.1162/003355397555299.

[24] F. Fang, Y. Xin, X. Yun und X. Haitao. "An opponent's negotiation behavior model to facilitate buyer-seller negotiations in supply chain management". In: *2008 International Symposium on Electronic Commerce and Security*. IEEE. 2008, S. 582–587. DOI: 10.1109/ISECS.2008.93.

[25] S. S. Fatima, M. Wooldridge und N. R. Jennings. "A comparative study of game theoretic and evolutionary models of bargaining for software agents". In: *Artificial Intelligence Review* 23.2 (2005), S. 187–205. DOI: 10.1007/s10462-004-6391-1.

[26] S. Fatima, M. Wooldridge und N. R. Jennings. "Comparing equilibria for game theoretic and evolutionary bargaining models". In: *5th International Workshop on Agent-Mediated E-Commerce*. 2003, S. 70–77.

[27] J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel und I. Mordatch. "Learning with opponent-learning awareness". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents und Multiagent Systems. 2018, S. 122–130.

[28] J. Garcıa und F. Fernández. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16.1 (2015), S. 1437–1480.

[29] I. Goodfellow, Y. Bengio und A. Courville. *Deep learning*. MIT press, 2016.

[30] J. Gwak und K. M. Sim. "Bayesian learning based negotiation agents for supporting negotiation with incomplete information". In: *World Congress on Engineering 2012. July 4-6, 2012. London, UK*. Bd. 2188. International Association of Engineers. 2010, S. 163–168.

[31] V. Haberland, S. Miles und M. Luck. "Adaptive Negotiation for Resource Intensive Tasks in Grids." In: *STAIRS*. 2012, S. 125–136.

[32] G. Hardin. "The tragedy of the commons: the population problem has no technical solution; it requires a fundamental extension in morality." In: *science* 162.3859 (1968), S. 1243–1248. DOI: 10.1126/science.162.3859.1243.

[33] J. H. Holland. "Properties of the bucket brigade". In: *Proc. 1st International Conference on Genetic Algorithms, 1985*. L. Erlbaum Associates. 1985, S. 1–7.

[34] J. H. Holland. "Studying complex adaptive systems". In: *Journal of systems science and complexity* 19.1 (2006), S. 1–8.

[35] E. Hughes, J. Z. Leibo, M. Phillips, K. Tuyls, E. Dueñez-Guzman, A. G. Castañeda, I. Dunning, T. Zhu, K. McKee, R. Koster et al. "Inequity aversion improves cooperation in intertemporal social dilemmas". In: *Advances in neural information processing systems*. 2018, S. 3326–3336.

[36] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman et al. "Human-level performance in 3D multiplayer games with population-based reinforcement learning". In: *Science* 364.6443 (2019), S. 859–865. DOI: 10.1126/science.aau6249.

[37] M. A. Janssen, R. Holahan, A. Lee und E. Ostrom. "Lab experiments for the study of social-ecological systems". In: *Science* 328.5978 (2010), S. 613–617. DOI: 10.1126/science.1183532.

[38] J. Kisiala. "Conditional value-at-risk: Theory and applications". In: *arXiv preprint arXiv:1511.00140* (2015).

[39] P. Kollock. "Social dilemmas: The anatomy of cooperation". In: *Annual review of sociology* 24.1 (1998), S. 183–214. DOI: 10.1146/annurev.soc.24.1.183.

[40] S. S. Komorita. "Cooperative choice in decomposed social dilemmas". In: *Personality and Social Psychology Bulletin* 13.1 (1987), S. 53–63. DOI: 10.1177/0146167287131005.

[41] K. A. Konrad und F. Morath. "Bargaining with incomplete information: Evolutionary stability in finite populations". In: *Journal of Mathematical Economics* 65 (2016), S. 118–131. DOI: 10.2139/ssrn.2490351.

[42] S. Kraus. "Agents Contracting Tasks in Non-Colla Environments". In: *Proceedings of the eleventh national conference on Artificial intelligence*. 1993, S. 243–248.

[43] I. Kwee, M. Hutter und J. Schmidhuber. "Market-based reinforcement learning in partially observable worlds". In: *International Conference on Artificial Neural Networks*. Springer. 2001, S. 865–873. DOI: 10.1007/3-540-44668-0_120.

[44] G. J. Laurent, L. Matignon, L. Fort-Piat et al. "The world of Independent Learners is not Markovian". In: *International Journal of Knowledge-based and Intelligent Engineering Systems* 15.1 (2011), S. 55–64. DOI: 10.3233/KES-2010-0206.

[45] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki und T. Graepel. "Multi-Agent Reinforcement Learning in Sequential Social Dilemmas". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS. 2017, S. 464–473.

[46] A. Lerer und A. Peysakhovich. "Maintaining cooperation in complex social dilemmas using deep reinforcement learning". In: *arXiv preprint arXiv:1707.01068* (2017).

[47] M. L. Littman. “Markov games as a framework for multi-agent reinforcement learning”. In: *Machine learning proceedings 1994*. Elsevier, 1994, S. 157–163. DOI: 10.1016/B978-1-55860-335-6.50027-1.

[48] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel und I. Mordatch. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”. In: *Advances in Neural Information Processing Systems*. 2017, S. 6382–6393.

[49] A. Lupu und D. Precup. “Gifting in Multi-Agent Reinforcement Learning”. In: *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems*. 2020, S. 789–797.

[50] L. v. d. Maaten und G. Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), S. 2579–2605.

[51] J. E. Maki, D. M. Hoffman und R. A. Berk. “A time series analysis of the impact of a water conservation campaign”. In: *Evaluation Quarterly* 2.1 (1978), S. 107–118. DOI: 10.1177/0193841X7800200105.

[52] R. McFarlane. “A survey of exploration strategies in reinforcement learning”. In: *McGill University* (2018).

[53] D. Mguni, J. Jennings, S. V. Macua, E. Sison, S. Ceppi und E. M. De Cote. “Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems”. In: *arXiv preprint arXiv:1901.10923* (2019).

[54] M. S. Miller und K. E. Drexler. “Comparative ecology: A computational perspective”. In: *The Ecology of Computation. North-Holland* (1988).

[55] M. S. Miller und K. E. Drexler. “Markets and computation: Agoric open systems”. In: *The ecology of computation* 1 (1988).

[56] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al. “Human-Level Control through Deep Reinforcement Learning”. In: *Nature* 518.7540 (2015), S. 529. DOI: 10.1038/nature14236.

[57] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), S. 529–533. DOI: 10.1038/nature14236.

[58] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya und T. Tanaka. “Parametric return density estimation for reinforcement learning”. In: *arXiv preprint arXiv:1203.3497* (2012).

[59] J. F. Nash Jr. “The bargaining problem”. In: *Econometrica: Journal of the Econometric Society* (1950), S. 155–162. DOI: 10.1515/9781400884087-006.

[60] S. Omidshafiei, C. Papadimitriou, G. Piliouras, K. Tuyls, M. Rowland, J.-B. Lespiau, W. M. Czarnecki, M. Lanctot, J. Perolat und R. Munos. "α-rank: Multi-agent evaluation by evolution". In: *Scientific reports* 9.1 (2019), S. 1–29.

[61] M. J. Osborne und A. Rubinstein. *A course in game theory*. MIT press, 1994.

[62] E. Ostrom. *Governing the commons: The evolution of institutions for collective action*. Cambridge university press, 1990. DOI: 10.1017/CBO9781316423936.

[63] E. Ostrom, J. Walker und R. Gardner. "Covenants with and without a sword: Self-governance is possible". In: *The American Political Science Review* (1992), S. 404–417. DOI: 10.2307/1964229.

[64] L. Panait und S. Luke. "Cooperative Multi-Agent Learning: The State of the Art". In: *Autonomous agents and multi-agent systems* 11.3 (2005), S. 387–434. DOI: 10.1007/s10458-005-2631-2.

[65] I. V. Papaioannou, I. G. Roussaki und M. E. Anagnostou. "Neural networks against genetic algorithms for negotiating agent behaviour prediction". In: *Web Intelligence and Agent Systems: An International Journal* 6.2 (2008), S. 217–233. DOI: 10.3233/WIA-2008-0138.

[66] J. Perolat, J. Z. Leibo, V. Zambaldi, C. Beattie, K. Tuyls und T. Graepel. "A multi-agent reinforcement learning model of common-pool resource appropriation". In: *Advances in Neural Information Processing Systems*. 2017, S. 3643–3652.

[67] A. Peysakhovich und A. Lerer. "Prosocial learning agents solve generalized stag hunts better than selfish ones". In: *arXiv preprint arXiv:1709.02865* (2017).

[68] Z. Rahaie und H. Beigy. "Toward a solution to multi-agent credit assignment problem". In: *2009 International Conference of Soft Computing and Pattern Recognition*. IEEE. 2009, S. 563–568. DOI: 10.1109/SoCPaR.2009.112.

[69] D. G. Rand, C. E. Tarnita, H. Ohtsuki und M. A. Nowak. "Evolution of fairness in the one-shot anonymous Ultimatum Game". In: *Proceedings of the National Academy of Sciences* 110.7 (2013), S. 2581–2586.

[70] C. W. Reynolds. "Flocks, herds and schools: A distributed behavioral model". In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 1987, S. 25–34. DOI: 10.1145/280811.281008.

[71] R. T. Rockafellar, S. Uryasev et al. "Optimization of conditional value-at-risk". In: *Journal of risk* 2 (2000), S. 21–42. DOI: 10.21314/JOR.2000.038.

[72] A. E. Roth. *Axiomatic models of bargaining*. Bd. 170. Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-51570-5.

[73] A. Rubinstein. "Perfect equilibrium in a bargaining model". In: *Econometrica: Journal of the Econometric Society* (1982), S. 97–109. DOI: 10.2307/1912531.

[74] K. Schmid, L. Belzner, T. Gabor und T. Phan. "Action markets in deep multi-agent reinforcement learning". In: *International Conference on Artificial Neural Networks*. Springer. 2018, S. 240–249. DOI: 10.1007/978-3-030-01421-6_24.

[75] K. Schmid, L. Belzner und C. Linnhoff-Popien. "Learning to Penalize Other Learning Agents". In: *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press. 2021. DOI: 10.1162/isal_a_00369.

[76] K. Schmid, L. Belzner, R. Müller, J. Tochtermann und C. Linnhoff-Popien. "Stochastic Market Games". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Hrsg. von Z.-H. Zhou. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, S. 384–390. DOI: 10.24963/ijcai.2021/54. URL: https://doi.org/10.24963/ijcai.2021/54.

[77] K. Schmid, L. Belzner, T. Phan, T. Gabor und C. Linnhoff-Popien. "Multi-agent Reinforcement Learning for Bargaining under Risk and Asymmetric Information." In: *ICAART (1)*. 2020, S. 144–151. DOI: 10.5220/0008913901440151.

[78] K. Schmid, R. Müller, L. Belzner, J. Tochtermann und C. Linhoff-Popien. "Distributed Emergent Agreements with Deep Reinforcement Learning". In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2021, S. 1–8. DOI: 10.1109/IJCNN52387.2021.9533333.

[79] J. Schulman, S. Levine, P. Abbeel, M. Jordan und P. Moritz. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, S. 1889–1897.

[80] J. Schulman, F. Wolski, P. Dhariwal, A. Radford und O. Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[81] L. S. Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2.28 (1953), S. 307–317. DOI: 10.1515/9781400881970-018.

[82] Y. Shoham und K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511811654.

[83] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton et al. "Mastering the Game of Go without Human Knowledge". In: *Nature* 550.7676 (2017), S. 354. DOI: 10.1038/nature24270.

[84] A. Smith et al. "The wealth of nations". In: *University of Chicago Bookstore* (2005).

[85] R. Sutton und A. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN: 9780262039246. DOI: 10.1109/TNN.1998.712192. URL: https://books.google.de/books?id=6DKPtQEACAAJ.

[86] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru und R. Vicente. "Multiagent Cooperation and Competition with Deep Reinforcement Learning". In: *PloS one* 12.4 (2017), e0172395. DOI: 10.1371/journal.pone.0172395.

[87] M. Tan. "Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents". In: *Proceedings of the 10th International Conference on International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc. 1993, S. 330–337. DOI: 10.1016/B978-1-55860-307-3.50049-6.

[88] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782 (2019), S. 350–354. DOI: 10.1038/s41586-019-1724-z.

[89] N. Vlassis. "A concise introduction to multiagent systems and distributed artificial intelligence". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 1.1 (2007), S. 1–71. DOI: 10.2200/S00091ED1V01Y200705AIM002.

[90] J. X. Wang, E. Hughes, C. Fernando, W. M. Czarnecki, E. A. Duéñez-Guzmán und J. Z. Leibo. "Evolving intrinsic motivations for altruistic behavior". In: *arXiv preprint arXiv:1811.05931* (2018).

[91] W. Wang, J. Hao, Y. Wang und M. Taylor. "Towards Cooperation in Sequential Prisoner's Dilemmas: a Deep Multiagent Reinforcement Learning Approach". In: *arXiv preprint arXiv:1803.00162* (2018).

[92] C. J. Watkins und P. Dayan. "Q-Learning". In: *Machine learning* 8.3-4 (1992), S. 279–292. DOI: 10.1007/BF00992698.

[93] G. Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.

[94] R. Winett, J. Kagel, R. Battalio und R. Winkler. "The Effects of Monetary Rebates, Feedback, and Information on Residental Energy Consumption". In: *Journal of Applied Psychology* 63 (1978), S. 73–80.

[95] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun und S. Whiteson. "Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?" In: *arXiv preprint arXiv:2011.09533* (2020).

[96] J. Yang, A. Li, M. Farajtabar, P. Sunehag, E. Hughes und H. Zha. "Learning to incentivize other learning agents". In: *Advances in Neural Information Processing Systems* 33 (2020), S. 15208–15219.

[97] G. Zlotkin und J. S. Rosenschein. "Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains." In: *IJCAI*. Bd. 89. Citeseer. 1989, S. 20–25.