

Endbericht der „PG 474: Neue Ansätze für das Gefangenendilemma“

Bastian Baranski, Rüdiger Ehlers, Thusinthan Kajendran,
Björn Kosslers, Tomasz Polaszek, Ralf Reimholz,
Jens Schmidt, Danny Seis, Rafael Slodzinski,
Simon Steeg, Nils Wiemann, Marc Zimmermann

Betreuer:

Thomas Bartz-Beielstein, Fachbereich Informatik, LS 11
Karlheinz Schmitt, Fachbereich Informatik, LS 11
Jörn Mehnen, Fachbereich Maschinenbau, Institut für Spanende Fertigung

2. März 2006

Inhaltsverzeichnis

| | | |
|-----------|---|-----------|
| I | Einleitung | 7 |
| 1 | Vorwort | 9 |
| 2 | Motivation | 11 |
| 2.1 | Struktur der PG | 11 |
| 2.2 | Ziele einer PG | 11 |
| 2.3 | Ziele dieser PG | 12 |
| 2.3.1. | Formalisierung des Gefangenendilemmas | 12 |
| 2.3.2. | Anwendungen des Gefangenendilemmas | 13 |
| 2.4 | Motivation der PG-Teilnehmer | 14 |
| 3 | Organisation | 17 |
| 3.1 | Überblick | 17 |
| 3.2 | Formalitäten | 17 |
| 3.3 | Werkzeuge | 18 |
| II | Rückblick | 21 |
| 4 | Rückblick | 23 |
| 4.1 | Überblick | 23 |
| 4.2 | Einleitung, Motivation und Zielsetzung | 23 |
| 4.3 | Turnierumgebung und Regeln | 23 |
| 4.3.1. | Vorüberlegungen | 23 |
| 4.3.2. | Der Regeltext | 24 |
| 4.4 | Realisierung der Turnierumgebung | 27 |
| 4.5 | Prüfung der Einsendungen | 29 |
| 4.6 | Öffentlichkeitsarbeit um das Turnier und das Campusfest | 30 |
| 4.6.1. | Homepage | 30 |
| 4.6.2. | Handzettel | 30 |
| 4.6.3. | E-Mail | 30 |
| 4.6.4. | Radio | 32 |
| 4.6.5. | Poster | 32 |
| 4.6.6. | Sponsoren und Preise | 34 |
| 4.6.7. | Presse | 34 |
| 4.7 | Der Stand am Campusfest | 39 |
| 4.8 | Fazit | 40 |

| | | |
|------------|---|------------|
| 5 | Ergebnisse des Campusfestturniers | 41 |
| 5.1 | Klassifikation von Strategien | 41 |
| 5.1.1. | Harte Attribute | 41 |
| 5.1.2. | Weiche Attribute | 43 |
| 5.1.3. | Standardstrategien. | 44 |
| 5.1.4. | Strategien von PG-Teilnehmern | 45 |
| 5.1.5. | Strategien von Teilnehmern außerhalb der PG | 50 |
| 5.2 | Ergebnistabelle des Turniers | 63 |
| | | |
| III | Konferenzarbeiten | 67 |
| | | |
| 6 | Einleitung | 69 |
| | | |
| 7 | Alternative Turnierumgebungen | 71 |
| 7.1 | Einleitung | 71 |
| 7.2 | Zusammenfassung „Environments for the IPD“ | 72 |
| 7.2.1. | Die Problemstellung. | 72 |
| 7.2.2. | Neue Umgebungen. | 74 |
| 7.2.3. | Ergebnisse. | 75 |
| 7.2.4. | Analyse der Ergebnisse. | 77 |
| 7.3 | Weitere Ergebnisse | 80 |
| 7.3.1. | Gesamtergebnis | 80 |
| 7.3.2. | Entstehung der <i>Jerusalem</i> -Umgebungen | 83 |
| 7.3.3. | Ergebnisse der ökologischen Umgebung | 83 |
| 7.3.4. | Erkennungssequenzen. | 85 |
| 7.3.5. | Analyse der <i>Jerusalem over all</i> -Umgebung | 86 |
| 7.4 | Zusammenfassung | 94 |
| | | |
| 8 | Simulationen durch Softwareagenten | 95 |
| 8.1 | Einleitung | 95 |
| 8.2 | Axelrods Viscosity-Model | 95 |
| 8.3 | Beschreibung der entwickelten Software | 96 |
| 8.4 | Zusammenfassung „High Order Punishment“ | 98 |
| 8.4.1. | Motivation. | 98 |
| 8.4.2. | Erweitertes Simulationsmodell | 98 |
| 8.4.3. | Durchführung der Simulation | 99 |
| 8.4.4. | Ergebnisse und Analyse | 99 |
| 8.5 | Zusammenfassung „The Impact of Group-Reputation“ | 101 |
| 8.6 | Zusammenfassung | 103 |
| 8.6.1. | High-Order Punishment | 103 |
| 8.6.2. | Impact of Group-Reputation. | 103 |
| 8.6.3. | Softwareagenten und das IPD im Rahmen der Projektgruppe | 103 |
| | | |
| 9 | Strategiegewinnung durch EA | 105 |
| 9.1 | Einleitung | 105 |
| 9.1.1. | EA | 105 |
| 9.1.2. | Internes Turnier. | 105 |
| 9.2 | Beschreibung der Ansätze | 105 |
| 9.2.1. | LGP-Ansatz | 105 |

| | |
|---|------------|
| 9.2.2. ES-Ansatz | 108 |
| 9.3 Umgebungen: anpassend vs. koevolutiv | 114 |
| 9.3.1. Anpassende Evolution | 114 |
| 9.3.2. Koevolution | 114 |
| 9.4 Versuchsvorbereitungen | 115 |
| 9.4.1. Parameteroptimierung | 115 |
| 9.4.2. Strategiegewinnung und Entscheidungsbäume | 118 |
| 9.4.3. Trainings- und Testmenge | 121 |
| 9.5 Ergebnisse | 123 |
| 9.5.1. Anpassende Evolution | 123 |
| 9.5.2. Koevolution | 124 |
| 9.6 Analyse | 128 |
| 9.6.1. Anpassende Evolution | 128 |
| 9.6.2. Koevolution | 129 |
| 9.7 Zusammenfassung | 129 |
| 10 Fazit | 133 |
| IV Zusammenfassung | 135 |
| 11 Zusammenfassung | 137 |
| V Anhang | 141 |
| A Environments for the IPD dealing with the Master-Slave Problem | 143 |
| B High-Order Punishment and the Evolution of Cooperation | 153 |
| C The Impact of Group-Reputation in a Multi-Agent Environment | 161 |
| D Literaturverzeichnis | 171 |

Teil I.
Einleitung

1. Vorwort

Das vorliegende Dokument stellt das Ergebnis einer einjährigen Gruppenarbeit dar, welche von zwölf Studenten am Fachbereich Informatik der Universität Dortmund durchgeführt wurde. Diese Projektgruppe, im Folgenden als PG abgekürzt, wurde von zwei Betreuern aus der Informatik, LS 11 und einem Betreuer aus dem Fachbereich Maschinenbau begleitet. In den nächsten beiden Kapiteln wird ausführlicher auf die Organisation und Durchführung sowohl einer PG im Allgemeinen als auch dieser PG im Speziellen eingegangen. Diese PG beschäftigte sich mit dem iterierten Gefangenendilemma (*Iterated Prisoner's Dilemma*, IPD), einem Problem aus der Spieltheorie. Bei dem originalen Gefangenendilemma werden zwei Spieler gleichzeitig vor die Wahl gestellt, mit dem anderen zusammen zu arbeiten oder gegen ihn zu arbeiten. Die Punkte, die beide erhalten, sobald die Wahl jedes der beiden Spieler feststeht, richten sich sowohl nach der eigenen Antwort wie auch nach der Antwort des Gegners. Das iterierte Gefangenendilemma hat dieselben Regeln, erstreckt sich jedoch über eine für beide Spieler unbekannt Anzahl von Runden. Die exakte Formalisierung, die Auszahlungstabelle für die Punkte und eine nähere Beschreibung wird im nächsten Kapitel ausführlich dargelegt.

Im Verlauf des ersten Semesters wurde vornehmlich versucht, einen Überblick über bisherige Arbeiten zu dem Thema zu erlangen. Zu diesem Zweck veranstaltete die PG ein öffentliches Turnier, bei dem jeder Interessierte seine Strategien einsenden konnte. Die Art der Durchführung des Turniers, die Ergebnisse, die Analyse der Ergebnisse und weitere Punkte können in dem Zwischenbericht der Gruppe nachgelesen werden. Er wurde als Download unter www.ipd-dortmund.de im pdf-Format zugänglich gemacht. Im zweiten Teil dieses Endberichts wird ein kurzer Überblick über das gegeben, was die PG im ersten Semester durchgeführt hat. Dort wird auch das Campus-Turnier noch einmal genauer erwähnt.

Das Ziel dieser PG am Anfang des zweiten Semesters war es, das Gefangenendilemma von verschiedenen Seiten zu beleuchten, und es unter mehreren Gesichtspunkten zu analysieren. Vieles, das im ersten Semester noch nicht geschafft wurde, wurde im zweiten Semester in Angriff genommen. Natürlich kann eine Gruppe nur eine begrenzte Anzahl an Themen untersuchen, jedoch zeigte sich, dass die Gruppe viele, teils sehr verschiedene Interessen hatte. Aus diesem Grund wurden drei Untergruppen gebildet, welche sich mit den verschiedenen Teilgebieten des IPD beschäftigten. Dabei haben sich alle Gruppen bemüht, die Ergebnisse der jeweils anderen Gruppen sowohl zu überprüfen, wie auch zu hinterfragen und natürlich in die eigene Arbeit einzubauen. So sollten Synergieeffekte erzielt werden, um sich damit gegenseitig zu unterstützen. Im dritten Teil dieses Endberichts werden die Arbeiten der einzelnen Untergruppen vorgestellt.

Die entscheidende Fragestellung der ersten Gruppe war, wie in einer anderen Turnierumgebung der Vorteil von Master-Slave-Strategien ausgeschaltet oder zumindest minimiert werden kann. Sie hat in diesem Zusammenhang eine von Axelrod entworfene Umgebung in verschiedenen Varianten getestet und darüber hinaus selber eine neue Umgebung entwickelt. Beide Umgebungen wurden auf ihr Verhalten bezüglich Master-Slave-Strategien hin untersucht, also in wie weit sich die verwendete Umgebung eignete, um den Master-Slave-Vorteil zu minimieren.

Die zweite Gruppe beschäftigte sich mit zwei verschiedenen Fragestellungen. Beide Ansätze basierten auf agentenbasierten Simulationen, welche Axelrods in seinem Paper [AH03] verwandte. Im ersten Ansatz sollte herausgefunden werden, wie sich das Konzept der Bestrafung auf eine agentenbasierte Simulation auswirkt. Im zweiten Ansatz wurde untersucht welche Auswirkungen eine Reputation der Agenten auf die Entwicklung der Kooperation hat.

Die dritte Untergruppe schließlich ging der Frage nach, ob und wenn ja, welche Vorteile es hätte,

die Entwicklung neuer Strategien für das IPD mit Hilfe von evolutionären Algorithmen vorzunehmen. Auch ist der Frage nachgegangen worden, wie man andere öffentliche Turniere dahingehend auswerten kann, dass man die beteiligten Strategien möglichst effizient nachbilden kann. Hierzu verwendete sie evolutionäre Algorithmen, welche Strategien als ganzes „erfanden“ und dann diese gefundenen Strategien evaluierten.

Weiterhin wurde beschlossen, vier Paper über die Ergebnisse der jeweiligen Arbeiten zu schreiben, welche bei verschiedenen Konferenzen eingereicht wurden. Dies ist auch die Erklärung für den etwas anderen Stil dieses Endberichts, da die Paper einen großen Teil der Arbeit dieser PG darstellen. Die eingereichten Paper sind, im englischen Original, im Anhang zu finden, die jeweiligen Paper werden im dritten Teil des Endberichts bei der jeweiligen Untergruppe noch einmal auf Deutsch zusammengefasst.

Um einen Überblick über den gesamten Endbericht zu gewährleisten, wird nun noch einmal die gesamte Struktur des Endberichts zusammengefasst. In den nächsten beiden Kapiteln der Einleitung wird zunächst die Struktur dieser PG erläutert, sowie das IPD formalisiert. Der zweite Teil des Endberichts soll einen kurzen Rückblick gewähren, was im letzten Semester getan wurde und unter welchen Voraussetzungen dieses Semester begann. Wie bereits erwähnt, stellen die drei Kapitel im dritten Teil dieses Endberichts die Arbeiten der unterschiedlichen Gruppen eingehender vor. Im letzten Teil des Endberichts wird dann ein Resümee über die beiden Semester gezogen und die Gruppe stellt dar, wo die Vor- und die Nachteile der spezifischen Struktur dieser PG lagen. Im Anhang schließlich finden sich die zu den Konferenzen eingereichten Paper.

2. Motivation

2.1. Struktur der PG

Eine Projektgruppe besteht aus einem oder mehreren Projektgruppenveranstaltern und 8-12 Projektgruppenteilnehmern. Der Veranstalter (bzw. zumindest einer der Veranstalter) ist Hochschullehrer oder wissenschaftlicher Mitarbeiter des Fachbereich Informatik. Die Teilnehmer sind Studierende mit Hauptfach Informatik im zweiten Studienabschnitt (d. h. mit abgeschlossenem Vordiplom).

Der Veranstalter wacht darüber, dass die sachlichen und zeitlichen Planungen der PG mit den angestrebten PG-Zielen (dies kann der Beweis einer Theorie sein, das Schreiben eines Programms mit vorgegebener Funktionsweise, das Lösen einer Problemstellung oder auch das Erarbeiten eines Algorithmus) weitgehend übereinstimmen.

Der Veranstalter ist bemüht, seine zunächst zwangsläufig leitende Funktion zugunsten einer mehr und mehr rein beratenden Funktion aufzugeben und somit die Teilnehmer der PG zu eigenverantwortlichem Planen und Arbeiten zu führen. Die Veranstalter nehmen bis zum Abschluss der PG an den PG-Diskussionen teil. Der Veranstalter versucht, etwaige Unstimmigkeiten möglichst durch Diskussionen im Rahmen der PG zu bereinigen.

Unsere Projektgruppe hat drei Projektgruppenveranstalter bzw. Betreuer: Thomas Bartz-Beielstein (Fachbereich Informatik, Lehrstuhl 11), Karlheinz Schmitt (Fachbereich Informatik, Lehrstuhl 11) und Jörn Mehnen (Fachbereich Maschinenbau, Institut für Spanende Fertigung).

Die studentischen Teilnehmer sind Bastian Baranski, Rüdiger Ehlers, Thusinthan Kajendran, Björn Kosslers, Tomasz Polaszek, Ralf Reimholz, Jens Schmidt, Danny Seis, Rafael Slodzinski, Simon Steeg, Nils Wiemann und Marc Zimmermann.

In diesem Kapitel soll nun näher auf die Motivationen eingegangen werden, sowohl auf die allgemeine Motivation einer PG, d. h. wieso und mit welcher Zielsetzung sie überhaupt durchgeführt wird, als auch auf die Motivation der Verantwortlichen für diese PG. Als letzter Punkt soll auf die Motivationen der einzelnen Teilnehmer, diese PG zu wählen, eingegangen werden.

2.2. Ziele einer PG

An der Universität Dortmund werden die Projektgruppen seit 1972 in jedem Semester durchgeführt. Projektgruppen finden in zwei jeweils aufeinander folgenden Semestern statt. Der Gesamtumfang einer Projektgruppe beträgt 14-16 Semesterwochenstunden (zwei Semester à sechs bis acht Semesterwochenstunden). Die tatsächliche Arbeitsbelastung des einzelnen Teilnehmers ist so zu bemessen, dass ein paralleler Besuch anderer Lehrveranstaltungen im Gesamtumfang von jeweils sechs bis zehn Semesterwochenstunden möglich ist. Die Motivationen, mit denen die PG an der Universität Dortmund ursprünglich eingeführt wurden, sind auch heute noch aktuell, wenn nicht gar aktueller denn je.

Zuerst sollen den Teilnehmern Fähigkeiten und Methoden vermittelt werden, welche sie in die Lage versetzt, sich in neue Aufgabenstellungen und Arbeitsgebiete einzuarbeiten, sowie neue Technologien und Programme anwenden zu können. Des Weiteren wird bezweckt, den einzelnen Teilnehmern ihre Teamfähigkeiten bewusst zu machen und diese zu steigern. Da viele Probleme zu groß und zu komplex sind, müssen sie in kleinere Teilprobleme zerlegt werden.

Die Zusammenführung und die Zusammenarbeit der Teilnehmer untereinander erfordert ein hohes Maß an so genannten „Softskills“, also Eigenschaften wie Kommunikationsfähigkeit, Bewältigung von Problemen in der Gruppe, Zurückstellen von rein persönlichen Zielen bzw. Vorteilen zum Wohle der Gruppe und Kompromissbereitschaft. Gerade in der heutigen Zeit, mit der Unsicherheit des Arbeitsmarktes und den schwindenden Stellenangeboten, stellt eine Praxiserfahrung einen nicht zu unterschätzenden Vorteil dar. Viele Firmen legen Wert darauf, dass ihre Mitarbeiter bereits in Teamarbeit und Projektarbeiten geübt sind. Die eben genannten Ziele kann keine Veranstaltung in Form einer Vorlesung oder eines Seminars bewältigen. Daher wurde die Notwendigkeit erkannt, eine neue Art der Veranstaltung anzubieten. Genau dies leistet die PG, allerdings nur in einem gewissen Rahmen. Daher kann die PG nur den Grundstock zu den oben genannten Fähigkeiten legen.

2.3. Ziele dieser PG

Die Hauptaufgabe dieser PG bestand ursprünglich darin, gute Strategien für das iterierte Gefangenendilemma zu entwickeln. Diese Strategien sollten auch mittels evolutionärer Verfahren gefunden werden. Das übergeordnete Ziel hierbei war eine erfolgreiche Teilnahme an einem der internationalen Wettbewerbe, die anlässlich des 20. Geburtstages des Gefangenendilemmas veranstaltet werden. Der erste fand bereits im Juli 2003 während der Konferenz „Congress of Evolutionary Computation“ statt. Die Hauptziele dieser PG waren:

1. Wiederhole das Originalexperiment.
2. Untersuche die Problemstellung unter Rauschen.
3. Entwickle eine Strategie für ein „multi-player“- und „multiple-choice“-IPD
4. Wiederhole das Originalexperiment unter der Voraussetzung, dass jeder Teilnehmer nur eine Strategie angeben darf.

Das Minimalziel war der konzeptionelle Entwurf, die Implementierung und Dokumentation von Algorithmen zum Auffinden einer guten Strategie für das IPD. Als Grundlage hierfür sollten existierende metaheuristische Verfahren herangezogen und verglichen werden.

Die Motivation, sich mit dem Gefangenendilemma auseinander zusetzen, liegt darin, dass viele kritische Situationen aus dem Bereich der Politik, Ökonomie oder Soziologie durch das Gefangenendilemma modelliert werden können. Um diese Modellierung vornehmen zu können, wird eine Formalisierung des Gefangenendilemma als ein Spiel benötigt.

2.3.1. Formalisierung des Gefangenendilemmas

Bei dem Gefangenendilemma (*Prisoner's Dilemma*, *PD*) werden zwei Verdächtige, die ein Verbrechen begangen haben, verhaftet und getrennt voneinander vernommen. Jeder der beiden hat zwei Möglichkeiten zur Auswahl: entweder er gesteht, oder er leugnet das Verbrechen. Falls nur ein Angeklagter gesteht und damit den anderen belastet, kommt er ohne Strafe davon, und der andere wird mit fünf Jahren Haft bestraft. Leugnen beide, werden sie jeweils für zwei Jahre eingesperrt. Gestehen beide, so erhalten sie eine Strafe von vier Jahren. Diese Situation wird in Tabelle 2.1 dargestellt.

Tabelle 2.1.: Bestrafungsmatrix für das Gefangenendilemma

| A/B | leugnen | gestehen |
|----------|---------|----------|
| leugnen | (2,2) | (5,0) |
| gestehen | (0,5) | (4,4) |

Wie anhand der Bestrafungsmatrix leicht zu erkennen ist, scheint ein Geständnis günstiger zu sein. Dies ist auch einleuchtend, denn gesteht ein Spieler, kann er im schlimmsten Falle für vier Jahre ins Gefängnis kommen, hat aber auch die Möglichkeit ohne Strafe davon zukommen. Leugnet er aber, so läuft er Gefahr, dass sein „Kontrahent“ gesteht und er somit für fünf Jahre ins Gefängnis kommt. Das Dilemma liegt nun darin, dass es für jeden Spieler unabhängig vom Verhalten des anderen, vorteilhafter ist zu gestehen, dass jedoch beidseitiges Gestehen für jeden Spieler ungünstiger ist als gemeinsames Leugnen [Axe05]. Das liegt daran, dass zwischen den Spielern keine verbindlichen Absprachen bzw. Verträge abgeschlossen werden dürfen bzw. können. Spieltheoretisch betrachtet ist es ein *nicht-kooperatives* Spiel, bei dem „gestehen“ die dominante Strategie beider Spieler ist.

Um das Gefangenendilemma besser analysieren zu können, wird die Matrix in eine Auszahlungsmatrix überführt. Hierbei wird die Strafe auf Punkte abgebildet, welche die Spieler erhalten. Die Wahlmöglichkeiten „leugnen“ und „gestehen“ wird in „kooperieren,“ und „defektieren“ umbenannt. Diese Transformationen ergeben die Auszahlungsmatrix in Tabelle 2.2.

Tabelle 2.2.: Auszahlungsmatrix nach der Transformation

| A/B | kooperieren | defektieren |
|-------------|-------------|-------------|
| kooperieren | (3,3) | (0,5) |
| defektieren | (5,0) | (1,1) |

Für das weitere Vorgehen werden zusätzliche Begriffe eingeführt [Axe05]:

- **Temptation (T)**: Versuchung zu defektieren
- **Reward (R)**: Belohnung für wechselseitige Kooperation
- **Punishment (P)**: Strafe für wechselseitige Defektion
- **Sucker's payoff (S)**: Auszahlung des gutgläubigen Opfers

Diese Werte können beliebig gewählt werden, solange die folgenden Bedingungen eingehalten werden.

- $T > R > P > S$
- $R > (T + S)/2$

In der Tabelle 2.2 wurden die Variablen wie folgt belegt: $T=5$, $R=3$, $P=1$ und $S=0$.

Das IPD entsteht durch die mehrfache Wiederholung des normalen Gefangenendilemmas. Das Problem wird durch den Faktor Zeit ergänzt und Spieler können auf die Entscheidungen ihrer Gegenspieler zu einem weitaus späteren Zeitpunkt durch Bestrafung (Defektion) oder Belohnung (Kooperation) reagieren. Hierbei werden zwei Varianten unterschieden. Bei der ersten Variante ist die Rundenanzahl festgelegt. Der Nachteil dieser Variante ist jedoch, dass durch die feste Rundenanzahl, die allen Spielern von vornherein bekannt ist, allgemein weniger Kooperation zwischen den Mitspielern entsteht. Bei der zweiten Variante ist die Rundenanzahl variabel, so dass die Spieler die tatsächlich zu spielende Rundenanzahl nicht kennen und dadurch Kooperation gefördert wird.

2.3.2. Anwendungen des Gefangenendilemmas

Ein Beispiel für eine Gefangenendilemma-Situation findet sich in der Werbeindustrie [Man01]. Man betrachte den Oligopolmarkt zweier namhafter US-Zigarettenhersteller und deren Ausgaben für Werbung im Fernsehen. Diese Situation lässt sich anhand der Auszahlungsmatrix in Tabelle 2.3 darstellen.

Tabelle 2.3.: Gewinnverteilung bei allen möglichen Konstellationen

| Hersteller A/ Hersteller B | Werbung | Keine Werbung |
|----------------------------|---------|---------------|
| Werbung | (1,1) | (18,0) |
| Keine Werbung | (0,18) | (10,10) |

Die Unternehmen haben jeweils zwei Möglichkeiten zur Auswahl: Entweder werben sie für ihr Produkt oder sie werben nicht für ihr Produkt. Die Strategie „Werbung“ entspricht der Defektion und „Keine Werbung“ kann als Kooperation beider Unternehmen aufgefasst werden. Wirbt keiner der beiden, so teilen sie sich den Markt und jeder erwirtschaftet einen Gewinn von insgesamt zehn Mio. US \$. Entschließen sich beide Unternehmen zu werben, so teilen sie sich weiterhin den Markt, der Gewinn wird jedoch durch die Werbekosten geschmälert, so dass dieser von zehn Mio. US \$ auf jeweils eine Mio. US \$ sinkt. Hierbei wird angenommen, dass Tabak-Werbung keine Nichtraucher akquiriert, sondern lediglich Raucher anderer Marken wechseln lässt. Betreibt hingegen nur ein Unternehmen Werbung, so verschieben sich die Marktanteile zum werbenden Unternehmen hin, worauf dieses seinen Gewinn erheblich steigern kann. Das nicht werbende Unternehmen geht leer aus. Wie beim Gefangenendilemma ist es für jedes Unternehmen besser, Werbung zu betreiben (defektieren) und einen geringeren Gewinn zu akzeptieren. Diese Situation herrschte bis in die frühen siebziger Jahre des letzten Jahrhunderts in den Vereinigten Staaten. Schließlich wurden beide Unternehmen ungewollt aus ihrem Dilemma befreit, als die US-Regierung ein Gesetz verabschiedete, welches Zigarettenwerbung im Fernsehen verbot. Überraschenderweise nutzen die Zigarettenhersteller ihren beachtlichen politischen Einfluss nicht aus, um das Gesetz zu verhindern. Dies lag daran, dass die von der Regierung aufgezwungene Kooperation beider Seiten entgegen kam, weil ihre Gewinne nicht mehr durch die Werbekosten geschmälert wurden.

2.4. Motivation der PG-Teilnehmer

Die vielen Teilnehmer haben aus unterschiedlichen Motiven heraus die Zusammenarbeit in dieser PG gewählt. Hier sollen nun einige der Motive genannt und erklärt werden.

Einer der meistgenannten Gründe sind die vielfältigen Anwendungsmöglichkeiten des IPD. Vornehmlich der psychologische Aspekt, also die Art und Weise der Auseinandersetzung mit dem Gegenüber, hat eine wichtige Rolle bei der Auswahl vieler Teilnehmer gespielt. Zu dieser Wahl passt auch das Nebenfach vieler Teilnehmer. Sie hoffen, durch das IPD die Informatik mit ihrem Nebenfach Psychologie verbinden zu können.

Weiterhin hat der hohe Anteil an Theorie eine sehr hohe Relevanz bei der Entscheidung der meisten Teilnehmer. Viele wollen nicht einfach nur vorgegebene Aufgaben programmieren, sondern auch selber forschen und entscheiden, in welche Richtung die PG sich bewegen soll. Diese Entscheidungsfreiheit stimmte und stimmt die meisten Teilnehmer, wenn nicht gar alle, sehr positiv.

Diese „ungenau“ Ausrichtung war ein weiterer Faktor bei der Wahl der PG, da die Teilnehmer den Kurs der PG zu einem großen Teil mitbestimmen können. Nicht nur bei der Ausrichtung und den genauen Zielen der PG, sondern vor allem auch bei der Organisation der einzelnen Treffen wie auch den Aufgabenstellungen und -verteilungen dürfen die Teilnehmer selbst entscheiden.

Auch sollen hier die Betreuer genannt werden. Diese haben während der Informationsveranstaltungen zur PG-Wahl ein positives Bild dieser PG gezeigt und damit sowohl das eigentliche Thema interessant gemacht, wie auch die damit zusammenhängenden Themen aufgezeigt. Dabei haben sie das Interesse einiger Studierender durch Stichworte wie „evolutionäre Algorithmen“, „von der Natur lernen“, „Ethnozentrismus“, etc. geweckt.

Als letzter aber wichtiger Punkt gilt für viele Studenten dieser PG die Möglichkeit, an Veröffentlichungen und Konferenzen mitzuarbeiten. Für die Mehrzahl der Teilnehmer ist dies die erste

Möglichkeit, mit einem internationalen Umfeld in Kontakt zu kommen und selbst Kontakte zu knüpfen.

3. Organisation

3.1. Überblick

Im ersten Semester hat sich die PG kontinuierlich zweimal pro Woche getroffen. Im zweiten Semester kamen zusätzlich Kleingruppensitzungen hinzu, um Probleme, die bei einem speziellen Thema auftraten, zu besprechen. Für alle Teilnehmer der Projektgruppe galt Anwesenheitspflicht, allerdings hatte jeder Teilnehmer aber die Chance, einmal pro Semester einen so genannten „Joker“ einzusetzen, um im Ausnahmefall für einen wichtigen Termin der Sitzung fernzubleiben. Das war jedoch nur möglich, wenn die PG geschlossen zustimmte.

Die PG-Sitzungen hatten den Zweck, in der Gruppe über Probleme, mögliche Lösungen und einzuschlagende Lösungswege diskutieren zu können. Während im ersten Semester die Organisation des öffentlichen IPD-Turniers, der Campusfestauftritt und Implementation einer Referenzturnierumgebung im Vordergrund standen, wurden die Schwerpunkte im zweiten Semester auf spezifische Probleme zum Gefangenendilemma gelegt.

Hierbei haben sich drei Hauptbereiche gebildet, die dann in Kleingruppen (drei bis sechs Personen) behandelt wurden. Diese Kleingruppen haben im zweiten Semester Paper erstellt und trafen sich dabei zusätzlich zu den Gruppensitzungen in ihren zugewiesenen Seminarräumen. Ergebnisse wurden dabei immer in Form von Präsentationen in den Gruppensitzungen vorgestellt. Insgesamt entwickelte sich im Laufe der Zeit eine Eigendynamik, nicht zuletzt begünstigt durch die Seminare in Bommerholz, die einiges dazu beigetragen haben, sich gegenseitig kennen zu lernen und effektiv miteinander zu arbeiten.

Wenn zu verschiedenen Themen weitere Informationen benötigt wurden oder Arbeiten nicht von der ganzen PG erledigt werden sollten, wurden zusätzliche kleine Arbeitsgruppen ins Leben gerufen. Für die Literaturbeschaffung zu bestimmten Themen, die Zuständigkeit für Sponsorenfragen des öffentlichen Turniers, später aber auch für die Struktur des Zwischen- und Endberichts und des abschließenden PG-Vortrages wurden Untergruppen gegründet. Falls die PG zu einem Zeitpunkt einen Überblick über ein bestimmtes Thema benötigte, wurden zudem PG-interne Vorträge gehalten, so dass sich nicht jeder PG-Teilnehmer dort separat einarbeiten musste.

3.2. Formalitäten

Durch die Erstellung von Protokollen wurden die Diskussionsergebnisse jeder Sitzung festgehalten und grafisch so aufbereitet, dass ersichtlich war, welche Aufgaben mit welcher Priorität noch zu erledigen waren. Dadurch wurde der sachliche Fortschritt der Arbeiten, die Einteilung in kleinere Arbeitsgruppen und die Einhaltung des Terminplans überwacht. Diese Protokolle wurden zentral in einer für alle einsehbaren und veränderbaren Wiki-Umgebung verwaltet. Auch die Kommunikation innerhalb der einzelnen Arbeitsgruppen und die über die PG-Sitzungen hinausgehenden Diskussionen wurden, wenn nicht persönlich, in der Wiki oder per E-Mail durchgeführt.

Jede Sitzung wurde durch einen Sitzungsleiter moderiert, dessen Entscheidungen, Abstimmungen und Zusammenfassungen vom Protokollanten festgehalten wurden. Beide Ämter wurden nach alphabetischer Reihenfolge vergeben, so dass insgesamt jeder Teilnehmer vier bis fünf Mal in der gesamten Zeit Sitzungsleiter war. Ziel dabei war es, die Moderation einer Sitzung zu erlernen. Dabei hat es sehr geholfen, dass Betreuer und PG-Teilnehmer den Sitzungsleiter nach Ende der Sitzung bewerteten und

sowohl positive als auch negative Punkte anführten. Diese Bewertung war auch immer konstruktiv und nicht destruktiv.

3.3. Werkzeuge

Zur Verfügung stand für jeden Teilnehmer ein Account in einem Rechnerpool, der aus Rechnern mit dem Betriebssystem *Linux* bestand. Bei den Arbeiten der Kleingruppen wurde auch häufig auf eigene Betriebssysteme, meistens *Windows*, zurückgegriffen. Zentral für die Kommunikation innerhalb der Kleingruppen und der PG insgesamt war ein dafür eingesetztes Wiki, eine Kommunikationsplattform, in der jeder Benutzer komfortabel Seiten selbst strukturieren und Dateien zur Verfügung stellen konnte. Geeignet wurde sich dabei auf *PmWiki 2.0*, die im Gegensatz zu manchen anderen Wikis keine Probleme beim Erstellen des Inhaltsverzeichnis hatte und für unsere Zwecke völlig ausreichte. Um Projekte wie die Implementierung der Turniersoftware oder auch Erstellung von End- und Zwischenbericht als auch der Paper zu organisieren, wurden CVS-Repositories eingerichtet. Somit konnte jederzeit mit dem Protokoll *SSH* auch von außerhalb auf die Repositories zugegriffen werden.

Dokumente wurden alle in dem frei verfügbaren Dokumentsatzsystem \LaTeX erstellt, u. a. der Zwischen- und Endbericht sowie alle Paper. Benutzt wurde hier unter Linux das vorinstallierte Latex-System und unter Windows die *MikTeX*-Distribution in Verbindung mit dem kostenlosen Editor *TeXnicCenter* 1.6.3beta. Es wurden Literatur-Bibliotheken im *BiBTeX*-Format erstellt und benutzt. Dazu wurde auf das Tool *JabRef* 1.8.1 zurückgegriffen, das ein grafisches Editieren von Literaturangaben im *BiBTeX*-Format möglich macht. Der Abschlussvortrag der PG wurde in \LaTeX mit Hilfe der *Beamer*-Klasse erstellt, die Präsentationen in \LaTeX komfortabel ermöglicht.

Für große Rechenaufgaben konnte das lehrstuhlinterne Batch-System *LSF* 6.1 leider nur eingeschränkt benutzt werden. Im ersten Semester wurde dieses kaum gebraucht, dafür umso mehr im zweiten Semester. Dort allerdings war das Batch-System aufgrund von Ausfällen und fehlenden Berechtigungen, die teilweise durch den Umzug des Lehrstuhls bedingt waren, nur sehr eingeschränkt benutzbar.

Um die Turnierumgebung in Software zu realisieren, entschied die Projektgruppe bei der Implementierung die Programmiersprache *Java* 1.5.0 einzusetzen. Für die Wahl dieser Programmiersprache waren zwei Faktoren ausschlaggebend. Erstens wird die Sprache schon seit einigen Jahren in den Grundstudiumsveranstaltungen an der Universität Dortmund zu Lehrzwecken eingesetzt. Es war anzunehmen, dass alle PG-Teilnehmer mit der Programmiersprache vertraut sind, wodurch die Implementierung einer Idee schnell und problemlos bewerkstelligt werden kann. Zweitens ist die Sprache plattformunabhängig, so dass die Turnierumgebung auf nahezu allen Architekturen ausführbar ist. Dabei wurden die Editoren *Eclipse* 3.1 und *NetBeans* 5.0 benutzt. Es kamen verschiedene Java-Bibliotheken zum Einsatz:

- *JUnit* 3.8.1, ein Framework zum Testen von Java-Programmen. Benutzt wurde dieses zum Testen unserer Turnierumgebung und zum Sicherstellen, dass die Standardstrategien auch richtig implementiert wurden.
- *HSQLDB* 1.8, eine Java-Datenbank.
- *WEKA* 3.4.6, eine Java-Bibliothek für Data Mining. Benutzt wurde diese für die Generierung von Entscheidungsbäumen für Strategien.
- *Apache POI* 2.5.1, eine Java-Bibliothek zum Schreiben und Lesen von verschiedenen Dateitypen, insbesondere Excel-Tabellen.

Für Berechnungen kamen noch das Statistik-Tool *R* 2.2.0 und *Matlab* 6 zum Einsatz. Beide dienten der Berechnung von Regressionsbäumen, wobei *Matlab* auch für die Erstellung von Graphen benutzt wurde. Als freie Alternative für *Matlab* wurde *Octave* 2.1.5.0 benutzt.

Teil II.
Rückblick

4. Rückblick

4.1. Überblick

In diesem Kapitel wird die Arbeit des ersten Semesters der Projektgruppe dargestellt. In dieser Zeit hat die PG ein öffentliches Turnier zum Gefangenendilemma veranstaltet. An dem Turnier gab es eine rege Beteiligung vieler Personen, wohl auch auf Grund der Preise, die dank verschiedener Sponsoren ausgeschrieben wurden. Das Hauptaugenmerk lag dabei vornehmlich auf den eingesandten Strategien. Es wurde viel Sorgfalt auf Klarheit bezüglich der Formulierungen gelegt. So wurden die Teilnahmebedingungen mehrmals umgeschrieben und korrigiert, auch die Umgebungsdefinitionen wurden exakt festgelegt. Dadurch war die PG im zweiten Semester in der Lage, viele der Themen, die bearbeitet wurden, wesentlich gezielter anzugehen, als es ohne diese Vorarbeiten möglich gewesen wäre.

Wurde im zweiten Semester das IPD unter verschiedenen Gesichtspunkten untersucht, bestand im ersten Semester doch eher eine allgemeine Fragestellung. Es wurden Strategien entwickelt, Umgebungen getestet, Literatur gesichtet und online Recherche betrieben.

4.2. Einleitung, Motivation und Zielsetzung

Um das IPD anhand von Mitteln aus der Informatik lösen zu können, ist eine spieltheoretische Herangehensweise naheliegend. Das Spiel zweier Strategien gegeneinander ist leicht zu simulieren, der Ausgang oft sogar vorhersagbar. Sollen „allgemeingültige“ Aussagen über die Güte einer Strategie getroffen werden, so ist ein Turnier in einer sorgfältig gewählten Turnierumgebung unvermeidbar. Wie schon im ersten Teil beschrieben, war ein Ziel der PG, das Originalexperiment von Axelrod zu wiederholen. Um nun eine möglichst ausgeglichene und objektive Menge von Strategien zu erhalten, schien ein öffentliches Turnier eine gute Lösung zu sein. Zeitpunkt der Durchführung war das Campusfest der Universität Dortmund. Die Teilnehmer sollten einem verbindlichen Regelwerk folgen und konnten mit einer guten Strategie hochwertige Preise gewinnen. Diese Preise wurden mit Geldern des Lehrstuhls und des Dekanats, aber auch durch andere Sponsoren finanziert. Um eine möglichst große Zahl von Einsendungen zu erhalten, wurden zudem vielfältige Werbemaßnahmen eingesetzt, die im Folgenden benannt und beschrieben werden. Ebenso musste eine Software entwickelt werden, die das Turnier mit den eingesandten Strategien durchführen konnte. Damit sich die eingesandten Strategien nicht zu sehr ähnelten, sollte das Turnier möglichst viele unterschiedliche Teilnehmer anlocken. Ergebnis des Turniers für die PG sollte eine möglichst große Datensammlung sein. Anhand dieser Datensammlung sollten Aussagen über die Eigenschaften guter oder schlechter Strategien in der verwendeten Turnierumgebung formuliert werden können.

4.3. Turnierumgebung und Regeln

4.3.1. Vorüberlegungen

Kernstück eines Turniers sind die zugrundeliegenden Regeln und die Umgebung, in der das Turnier stattfindet. Das Beispiel der Gruppe „Southampton“ hat gezeigt, dass es für manche Turnierumgebungen „bessere“ Strategien gibt als die bisher allgemein bekannten. Bei der Entwicklung von sinnvollen Regeln waren schon vorab einige Punkte klar:

- Eine Kooperation zwischen Strategien (im Sinne der so genannten „Southampton“- oder Master-Slave-Strategien; also Kommunikation über die Spielzüge) sollte erlaubt sein, aber nicht durch eine Masse von Slave-Strategien spielentscheidend wirken. Deshalb durfte jeder Teilnehmer nur maximal zwei Strategien einreichen. Eine andere Form der Kommunikation zwischen Strategien sowie die Speicherung von Zügen oder Ergebnissen über ein Spiel hinaus wurde untersagt.
- Teilnehmer, welche nur eine Strategie entwickelten, traten mit zwei Instanzen dieser einen Strategie beim Turnier an.
- Strategien der PG-Teilnehmer, sowie die zufällige arbeitende Strategie *Random* sollten auch zugelassen werden, wurden aber von der Gewinnverteilung ausgeschlossen.
- Außerdem wurden in den Regeln Formalitäten wie die Anzahl der Runden und Spiele, der Inhalt der Auszahlungsmatrix und der weitere Ablauf des Turniers (z. B. Verfahren bei Punktgleichheit zweier Strategien) festgelegt.

Die meisten Feinheiten der Regeln ergaben sich aus der Vorgabe, wie die Software zu verwenden bzw. anzusprechen sei, und einigen Formulierungen, die juristische Konsequenzen verhindern sollten. Zusätzlich gab es einen Passus zum Urheberrecht der eingesandten Strategien, welche u. U. zu Forschungszwecken weiterverwendet werden sollten. Es folgt der exakte Wortlaut der Regeln.

4.3.2. Der Regeltext

4.3.2.1. Teilnahme und Gewinne

- Jede Person, die diese Regeln anerkennt und sich entsprechend verhält, ist teilnahmeberechtigt. Minderjährige Personen müssen vor der Teilnahme einen Erziehungsberechtigten um Erlaubnis bitten.
- Jeder Teilnehmer darf eine oder zwei Strategien einsenden. Wird nur eine Strategie eingereicht, so wird diese doppelt instanziiert, d. h. sie tritt im Turnier zweimal an.
- Es werden nur Strategien zugelassen, die eine Nettorechenzeit von höchstens einer Sekunde pro Zug benötigen. Die Turnierleitung behält sich vor, Strategien, die mehr Zeit pro Zug benötigen, vom Turnier auszuschließen.
- Sendet ein Teilnehmer zwei Strategien ein, von denen jedoch eine ausgeschlossen werden muss, so nimmt die andere Strategie am Turnier teil, als hätte der Teilnehmer nur die eine, gültige Strategie eingesandt.
- Jeder Teilnehmer kann höchstens einen Preis gewinnen. Sollten beiden Strategien oder beide Klone einer Strategie einen Gewinnplatz erreichen, so gewinnt der Teilnehmer den Preis für die höhere Platzierung und alle Preise ab der zweiten Platzierung werden automatisch an den nächsten Platz weitergereicht.
- Die Turnierleitung behält sich vor, eigene Strategien sowie Computerstrategien, wie z. B. *Random*, am Turnier teilnehmen zu lassen. Diese Strategien sind aber von der Gewinnverteilung ausgeschlossen. Sollte eine dieser Strategien in der Rangliste einen Gewinnplatz erreichen, so wird dieser Preis mit allen folgenden automatisch an den jeweils nächsten Platz in der Rangliste weitergereicht.

4.3.2.2. Strategie

- Die Strategien sollten als Java-Klassen, die das Interface „Strategy“ implementieren, bereitgestellt werden. Sie müssen dann folgende Methoden realisieren:
 - **Object clone()**: erzeugt einen Klon der Strategie und gibt diesen zurück.
 - **void reset()**: setzt die internen Variablen auf die Anfangswerte. Es dürfen zwischen den Spielen KEINE Informationen gespeichert werden, alle Variablen müssen hier mit einem Wert initialisiert werden.
 - **String getName()**: gibt den Namen der Strategie zurück.
 - **int getFirstMove()**: Methode, die den ersten Zug der Strategie zurück gibt, wobei nur die beiden Werte 0 (*kooperieren*) und 1 (*defektieren*, sprich: *verraten*) als Rückgabewert zugelassen sind.
 - **int getMove(int lastOpponentMove)**: Methode, die den nächsten Zug zurückgibt, wobei als Argument der letzte Zug des Gegners übergeben wird. Auch hier sind nur die beiden Werte 0 und 1 zulässig.
- Strategien, die als Java-Klassen eingereicht werden, dürfen keine so genannten „Endlosschleifen“ enthalten. Solche Strategien werden aus dem Turnier ausgeschlossen.
- Strategien dürfen als Pseudo-Code eingereicht werden. Die Turnierleitung gibt jedoch keine Garantie für die korrekte Umsetzung bzw. die korrekte Funktion dieser Strategien.

4.3.2.3. Turnier

- Das Turnier wird in Form eines „*Round-Robin*“ Turniers durchgeführt. Das heißt: Jede Strategie spielt gegen alle anderen Strategien und auch gegen sich selbst, in drei verschiedenen Durchgängen.
- Die Anzahl der Runden in einem Durchgang wird zufällig ermittelt. Dabei gibt es eine Mindestrundenzahl von 100 Runden im ersten, 500 Runden im zweiten und 1000 Runden im dritten Durchgang. Vor jeder Folgerunde wird der Durchgang mit einer Wahrscheinlichkeit von $10/\text{Mindestrundenzahl}$ abgebrochen, also mit Wahrscheinlichkeiten von 0.1, 0.02 bzw. 0.01. Die konkreten Rundenzahlen werden vor dem Turnier ermittelt und sind für alle Spiele gleich. Sie sind keiner Strategie vorab bekannt, auch denen der Veranstalter nicht.

4.3.2.4. Punktezählung

- Die Auszahlungen bzw. Punkte für eine einzelne Runde sind durch folgende Werte bzw. folgende Auszahlungsmatrix festgelegt:
 - Temptation to defect = 5
 - Reward = 3
 - Punishment = 1
 - Sucker's Payoff = 0
- Die Gesamtpunktzahl einer Strategie für ein Turnier ist die Summe aller erzielten Punkte in allen Durchgängen.
- Um den Zufall gering zu halten, wird das gesamte Turnier zehn Mal durchgeführt und ein Mittelwert der Gesamtpunktzahlen einer Strategie gebildet. Dieser Mittelwert ist Maß für die Reihenfolge der Strategien in der Rangliste.

Tabelle 4.1.: Die Auszahlungsmatrix

| A/B | kooperieren | defektieren |
|-------------|-------------|-------------|
| kooperieren | (3,3) | (0,5) |
| defektieren | (5,0) | (1,1) |

- Sollte es mehrere Platzierungen auf einem der Gewinnplätze geben, so entscheidet ein Stichturnier der beiden Strategien zusammen mit der Strategie *Random*, die sich zufällig verhält, über die Platzierung. Dieses Stichturnier wird so oft durchgeführt, bis ein eindeutiges Ergebnis festliegt. Die andere Strategie rutscht entsprechend einen Platz in der Rangliste nach unten.
- Die Strategie mit den meisten Punkten erreicht den obersten Platz in der Rangliste.

4.3.2.5. Einsendeschluss

- Strategien können spätestens bis zum 15. Juni 2005 eingereicht werden. Es gilt der Eingangsstempel oder das Eingangsdatum bei E-Mails.

4.3.2.6. Hinweis zum Urheberrecht

- Das Ziel der Projektgruppe und damit auch des Turniers ist es u. a. mittels evolutionärer Verfahren eine Strategie zur Lösung des *iterierten Gefangenendilemmas* zu entwickeln, die internationalen Vergleichen standhält. Aus diesem Grund werden alle zum Turnier eingesandten Strategien analysiert und für Forschungszwecke genutzt. Dies könnte eventuell eine Analyse einer Strategie in öffentlich zugänglichen Publikationen umfassen. Mit der Teilnahme am Turnier stimmen die Teilnehmer etwaiger Verwendung der Strategien zu wissenschaftlichen Zwecken und der Verwendung in Publikationen in diesem Zusammenhang zu.
- Teilnehmer, die einer Verwendung ihrer Strategie zu wissenschaftlichen Zwecken oder der Veröffentlichung in damit verbundenen Publikationen nicht zustimmen, müssen sich mit Einsendung der Strategie schriftlich von der Verwendung durch diese Forschungsgruppe distanzieren. Die Verweigerung der Verwendung zu wissenschaftlichen Zwecken oder der Publikation entsprechender Ergebnisse verhindert nicht die Teilnahme am Turnier. Alle anderen Regeln bleiben von der Verweigerung unberührt und weiterhin gültig.

4.3.2.7. Sonstiges

- Die Turnierleitung behält sich vor, einzelne Teilnehmer unter Nennung berechtigter Gründe aus dem Turnier auszuschließen.
- Mit Einsendung einer Strategie werden diese Regeln anerkannt.
- Handlungen wider den Regeln führen zur Disqualifikation.
- Nachträglich entdeckte Regelverstöße eines Teilnehmers entziehen diesem etwaige Gewinnansprüche.
- Der Rechtsweg ist ausgeschlossen.

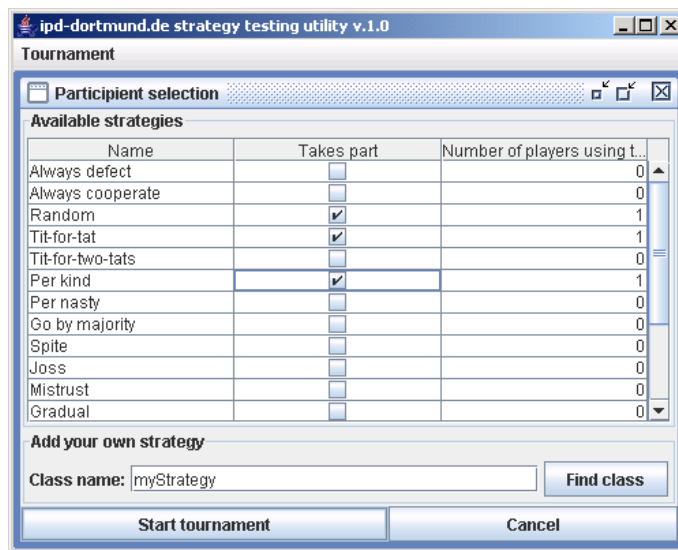


Abbildung 4.1.: Teilnehmerdialog zur Testumgebung

4.4. Realisierung der Turnierumgebung

Der konzeptionelle Entwurf der Turnierumgebung sah eine möglichst einfache und benutzerfreundliche Anwendung vor. Dies galt sowohl für die Benutzung der graphischen Oberfläche als auch für die Implementierung neuer Strategien und ihrer Einbindung in die Turnierumgebung. Dieser Punkt war wesentlich, da die Projektgruppe beschloss die Software allen Interessenten zur Verfügung zu stellen. Jeder Teilnehmer sollte die Möglichkeit erhalten, noch vor der Einsendung seine Strategie zu testen und ihre Funktionalität zu überprüfen. Hierzu wurde eine jar-Datei, „*Strategy Testing Utility*“ genannt, generiert, welche die Testumgebung enthielt. Diese wurde gemeinsam mit dem Regeltext und einer Anleitung zum Entwickeln eigener Strategien auf der Internetseite der Projektgruppe zum Download angeboten.

Die Testumgebung wird mit dem Befehl „*java -jar TestingUtility.jar*“ in der Kommandozeile aufgerufen. Gleich darauf erscheint das Hauptfenster der Anwendung.

Nun stehen dem Benutzer lediglich zwei Optionen zur Verfügung. Entweder beendet er die Anwendung oder startet ein neues Turnier. Jegliche Verwaltungsarbeit, wie zum Beispiel die Angabe einer Mindestundenzahl pro Durchgang oder einer Abbruchwahrscheinlichkeit wird dem Benutzer abgenommen und von der Applikation selbst durchgeführt. Ein neues Turnier wird über den Menüpunkt „*Tournament -> New*“ in der Menüleiste gestartet. Kurz darauf erscheint ein Dialogfenster, das erlaubt, teilnehmende Strategien für das Turnier zu bestimmen. Um dem Benutzer einen Vergleichsmaßstab zu geben wie gut oder schlecht seine Strategien in einer Turnierumgebung abschneiden könnten, entschied die Projektgruppe einige der gängigen Standardstrategien zum *iterierten Gefangenendilemma* als festen Bestandteil der Testsoftware zu implementieren. Abbildung 4.1 zeigt den Aufbau des Dialogfensters. Durch das Setzen eines Häkchens in der zweiten Spalte wird der Anwendung mitgeteilt, die genannte Strategie am Turnier teilnehmen zu lassen. Der Wert im Eingabefeld rechts bestimmt wie viele Instanzen der jeweiligen Strategie am Turnier teilnehmen. Wie bereits angesprochen sollte es auch möglich sein eigene Strategien in die Turnierumgebung einzubinden. Hierzu wurde in das untere Drittel des Dialogs ein Eingabefeld und ein Button integriert, welche diese Funktion realisieren. Hierbei genügt es den Klassennamen einer Strategie in das Textfeld einzutragen und anschließend den „*Find Class*“-Button zu drücken. Daraufhin wird die Strategie in die Liste verfügbarer Strategien aufgenommen. Zu beachten ist jedoch, dass die neuen Strategien bereits in einer kompilierten Form vorliegen und sich im selben Verzeichnis wie die Turnierapplikation befinden müs-

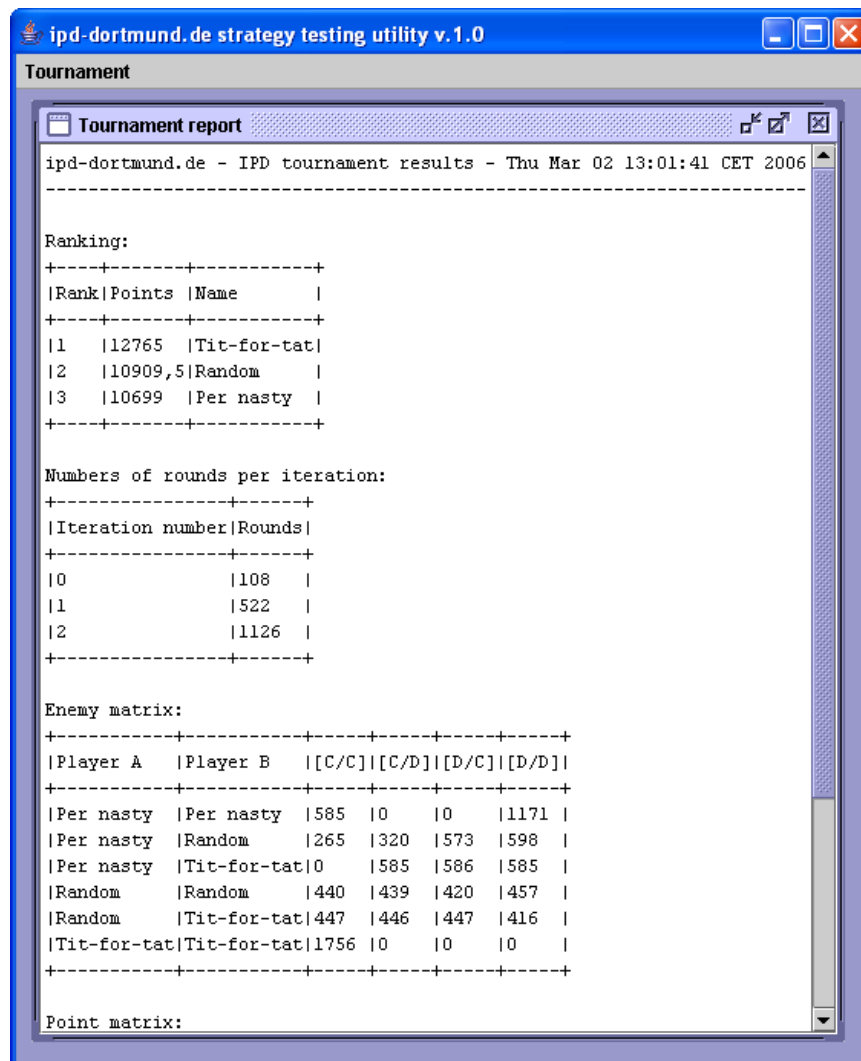


Abbildung 4.2.: Die Turnierergebnisse

sen. Die Berechnung des Turniers startet mit einem Klick auf den „Start tournament“-Button. Nach einer gewissen Berechnungszeit, die von der Anzahl der teilnehmenden Strategien und der Rundenanzahl pro Durchgang abhängt, wird das Turnierergebnis anhand verschiedener Matrizen (vgl. Abb. 4.2) präsentiert. Wie aus der Abbildung ersichtlich ist, generiert die Anwendung eine Bestenliste. Anhand dieser Tabelle kann für jede Strategie nachgeprüft werden wie erfolgreich sie war - also wie viele Punkte sie im gesamten Turnier gesammelt hat. Falls mehrere Instanzen einer Strategie am Turnier teilnehmen, erhält jede Instanz eine Nummer, die an den Strategienamen angehängt wird. Weiterhin zeigt der Turnierreport die Anzahl der Durchgänge des Turniers und die genaue Anzahl der gespielten Runden pro Durchgang an. Um eine ausführliche Analyse des Turniers durchführen zu können, generiert die Applikation zu guter Letzt die so genannte „Feindmatrix“ und die zugehörige „Punktematrix“. Die „Feindmatrix“ hält für jeweils zwei Strategien fest, wie oft sie im gesamten Turnier miteinander kooperiert haben, sich ausbeuten ließen oder gegenseitig defektierten. In der abschließenden „Punktematrix“ werden diese Ereignisse in Punkte für die jeweiligen Strategien umgerechnet. An dieser Stelle soll erwähnt werden, dass ein Kompromiss zwischen der Ausführlichkeit eines Turnierergebnisses und der dazu erforderlichen Hardware gefunden werden musste. So werden diese Matrizen ausschließlich generiert, falls weniger als 100 Strategien am Turnier teilnehmen.

4.5. Prüfung der Einsendungen

Ehe das eigentliche Turnier durchgeführt werden konnte, mussten alle Einsendungen geprüft werden. Die zur Überprüfung eingesetzten Kriterien konnten grob in zwei Gruppen unterteilt werden:

1. Formale Kriterien:

- Die Vorgaben aus den Regeln mussten eingehalten werden, z. B. das Löschen aller Variablen in der Reset-Methode, um die Speicherung von Daten über ein Spiel hinaus zu verhindern.
- Die Strategien mussten auf Endlosschleifen untersucht werden, oftmals durch Analyse des Kontrollflussgraphen. In den Regeln war dieser Punkt durch Vorgabe einer Zugzeit gesichert.

2. Inhaltliche Kriterien:

- Inhaltlich mussten die Strategien auf logische Fehler geprüft werden. So fiel ein „=“ statt einem „==“ bei der Ausführung nicht auf, es war jedoch deutlich, dass der Teilnehmer sich verschrieben hatte. Da viele Strategien kurz und einfach strukturiert waren, konnten solche Fehler in Kooperation mit dem jeweiligen Autor noch vor dem Turnier behoben werden.
- Es musste überprüft werden, ob eine Kooperation zwischen Strategien verschiedener Teilnehmer stattfindet, oder ob ein Teilnehmer unter verschiedenen Namen Einsendungen abgegeben hatte.

Obwohl die Regeln mehrfach von den meisten PG-Teilnehmern gegengelesen und auf Schlupflöcher untersucht wurden, gab es trotzdem Turnierteilnehmer, die versuchten, eine Kooperation zwischen mehr als zwei Strategien auszunutzen.

Dabei wurden von fünf verschiedenen Teilnehmern identische Slave-Strategien eingesandt, die auch mit gleicher Erkennungssequenz arbeiteten. Obwohl die Kooperation zwischen Strategien im Regeltext fahrlässigerweise nicht ausdrücklich verboten wurde, verstieß dieses Vorgehen gegen die Regel, dass ein Teilnehmer nur mit zwei Strategien antreten durfte. Da die Slave-Strategien nicht trivial programmiert waren und mit der gleichen Sequenz zur Erkennung arbeiteten, durfte angenommen werden, dass sie von einem Teilnehmer programmiert und an die anderen verteilt wurden, um durch eine größere Menge von Slave-Strategien Punkte zu erhalten. Der Verdacht wurde noch erhärtet als feststand, dass eine der Teilnehmerinnen nur den Slave eingesandt hatte, zu dem es keine passende Master-Strategie von ihr gab. Strategien abzugeben, die nur dafür programmiert wurden, bestimmten anderen Strategien von anderen Teilnehmern Punkte zu zuspieren, war nicht im Sinne des Turniers.

Dass dieser Fall in den Regeln nicht ausdrücklich, aber doch implizit verboten war, erschwerte die Entscheidung, wie mit diesem Fall umzugehen sei. Folgende Möglichkeiten wurden zur Diskussion gestellt:

- Alle Strategien könnten zugelassen werden, da die Kooperation zwischen den Strategien nicht explizit verboten war.
- Alle Strategien könnten disqualifiziert werden, da offensichtlich nicht jeder Teilnehmer eigene Strategien programmiert hatte und damit ein Teilnehmer mehr als eine Strategie eingereicht hatte. Dieser Fall konnte laut Regeltext zum Ausschluss aus dem Turnier führen.
- Die erste der eingesandten Strategien könnte zugelassen werden, alle anderen werden vom Turnier ausgeschlossen, da mindestens eine der Strategien nicht selbst geschrieben und von einem anderen Teilnehmer eingereicht wurde.

Die PG entschied sich aus Rücksicht auf die anderen Teilnehmer und wegen der nicht eindeutigen Formulierung in den Regeln für die dritte Möglichkeit.

4.6. Öffentlichkeitsarbeit um das Turnier und das Campusfest

Um einen möglichst großen Teilnehmerkreis für das Turnier zu gewinnen, griff die Projektgruppe auf verschiedene Mittel zurück. So wurde unter anderem das Internet und E-Mail als Medium zu Werbezwecken eingesetzt. Weiterhin bediente sich die Projektgruppe klassischer Medien wie Flyer und Radio, um Interessenten an der Universität Dortmund auf das anstehende Turnier aufmerksam zu machen. Neben diesen Aufgaben mussten ebenfalls noch Sponsoren angesprochen und gewonnen werden, um die angekündigten Preise für die besten Strategien finanzieren zu können.

4.6.1. Homepage

Die Idee eine Internetpräsenz aufzubauen kam während einer Diskussion zur Turniervorbereitung auf. Der Vorschlag wurde positiv aufgenommen, so dass zu diesem Punkt eine Untergruppe gebildet wurde. Zunächst bestand ihre Aufgabe darin, ein Konzept für den Aufbau der Seite zu erarbeiten und Vorschläge für die URL zu formulieren. Natürlich konnten die übrigen PG-Teilnehmer sie dabei mit eigenen Ideen und Anregungen unterstützen. Als Adresse der Webseite wurde die URL www.ipd-dortmund.de vorgeschlagen, welche die Projektgruppe auch annahm. Der erarbeitete Entwurf sah vor, die Webseite thematisch zu organisieren, so dass Teilnehmer des Turniers und Interessierte in der Lage sein sollten, die gewünschten Informationen schnell zu finden. Als Themengebiete kristallisierten sich zwei Bereiche heraus. Zum einen sollte die Projektgruppe selbst vorgestellt und ihre Arbeit zum Gefangenendilemma der Öffentlichkeit präsentiert werden. Hierzu gehört auch die Veröffentlichung neuer Erkenntnisse, welche im Rahmen der Projektgruppenarbeit gewonnen wurden. Zum anderen sollten alle Interessierten über die Webseite die Turnier-Software, die zugehörige Anleitung und den Regeltext beziehen können. Darüber hinaus fungierte die Webseite als offizielle Kontaktstelle bei Anfragen bezüglich des Turniers und bei technischen Problemen. Da sich die Projektgruppe dazu entschied, das Turnier auch international auszuschreiben, war und ist die Internetpräsenz sowohl in deutscher als auch in englischer Sprache verfügbar.

4.6.2. Handzettel

Die Flyer sind entworfen worden, um den Stand, der für das Campusfest beantragt wurde, bekannt zu machen und weitere Interessenten für das Turnier zu werben. Da alle PG-Teilnehmer der Meinung waren, dass diese Art der Werbung die Beste sei, wurden 2500 Flyer bei der Firma *FlyerWire* bestellt und gedruckt. Die Flyer wurden in sämtlichen Informatikvorlesungen, in den Mensen und am Campusfest verteilt bzw. ausgelegt. Die gesamten Kosten der Flyer übernahm die Krankenkasse AOK-Dortmund. Vorder- und Rückseite der Flyer sind in Abbildung 4.3 und Abbildung 4.4 zu sehen.

4.6.3. E-Mail

Mit der Ausarbeitung und dem Versenden einer E-Mail wurde eine Kleingruppe beauftragt. Sie sollte einen ansprechenden Text sowohl in deutscher als auch in englischer Sprache formulieren und zugleich eine Liste möglicher Ansprechpartner erstellen. Die E-Mail richtete sich hauptsächlich an Informatik-Fachschaften anderer Universitäten im deutschsprachigen Raum aber auch an die Initiative „Schulen ans Netz“, um Schüler und Lehrer für das Gefangenendilemma und das anstehende Turnier zu begeistern. Um internationales Publikum zu gewinnen, schrieb die Gruppe einige englischsprachige Informatik-Newsgruppen an und kontaktierte Graham Kendall, den Turnierveranstalter aus dem Jahr 2004. An dieser Stelle sei der Inhalt der deutschen E-Mail wiedergegeben:



Abbildung 4.3.: Die Vorderseite des Flyers

 **Das Gefangenendilemma**

Was? Spielt am Stand gegeneinander oder nimmt am Turnier teil und gewinnt! Turnierstrategien können online bis zum 15.06.05 eingesandt werden.

Wann? Samstag, den 18.06.05 von 10-16 Uhr, Tag der offenen Tür, Campusfest

Wo? Universität Dortmund, EF50, Stand der Projektgruppe "Gefangenendilemma"

...weitere Informationen unter www.IPD-dortmund.de

sponsored by...

Studenten-Service
www.unilife.de/wl

PPS MUSIC
Preisings Party Sound
WWW.ULF-PREISING.DE
Tel. 02307 - 97 000 77

Abbildung 4.4.: Die Rückseite des Flyers

Subject: *Wettbewerb zum iterierten Gefangenendilemma (IPD)*

Die Projektgruppe (PG) 474 der Universität Dortmund veranstaltet im Rahmen des Tages der offenen Tür am 18.6.2005 ein Turnier zum iterierten Gefangenendilemma (IPD). Sie sind hiermit herzlich eingeladen, an diesem teilzunehmen. Schicken Sie einfach bis zum 15.6.2005 eine Strategie an info@ipd-dortmund.de. Zu gewinnen gibt es einen iPod, mp3-Player und Memory-Sticks.

Teilnahmebedingungen und weitere Informationen erhalten Sie unter www.ipd-dortmund.de. Wir freuen uns schon auf Ihre Einsendungen.

Mit freundlichen Grüßen, Ihr Team der PG 474

4.6.4. Radio

Um eine möglichst hohe Teilnehmerzahl zu erreichen wurde als zusätzliches Medium für die Promotion des Turniers das Radio gewählt. Da die Universität Dortmund über einen eigenen Campus-Radiosender namens „eldorado*“ verfügt, schien es naheliegend mit den Verantwortlichen Kontakt aufzunehmen.

Ursprünglich hätte ein vorproduzierter Jingle im Tagesprogramm ausgestrahlt werden können. Nach Rücksprache mit den verantwortlichen Redaktionen für „*Musik*“ und „*Wort*“ wurde davon aber abgesehen, da die Problematik des Gefangenendilemmas für Nichteingeweihte nicht innerhalb der Zeitspanne eines Radio-Jingles von ca. 30 bis 60 Sekunden erklärbar wäre. Dies hätte einerseits Ideen bzw. die Teilnahme von Nicht-Experten ausgeschlossen, wäre andererseits der Chefredaktion des Radiosenders für das Tagesprogramm auch zu speziell gewesen.

Die nächste angedachte Variante war der Besuch einer moderierten Live-Sendung, in der Vertreter der Projektgruppe mit dem Moderator die Grundzüge des IPD besprochen und auf das Turnier hingewiesen hätten. Der Vorteil: es hätte mehr Zeit zur Verfügung gestanden und ein neutraler Moderator hätte Fragen stellen und beantworten lassen können, die auch ein Hörer stellen würde, der mit dem Thema nicht vertraut aber doch interessiert ist.

Aus organisatorischen Gründen auf Seiten von eldorado* kam diese Sendung nicht zustande. Als Kompromiss wurde statt des Live-Besuches im Radiostudio ein Telefoninterview aufgezeichnet, das am darauf folgenden Tag in einer moderierten Sendung ausgestrahlt wurde. Der Zeitrahmen dieses Interviews war selbstverständlich kleiner, reichte aber aus, um das Problem zu beschreiben und auf das Turnier und die Internetseite, auf der alle weiteren Informationen erhältlich waren, hinzuweisen.

4.6.5. Poster

Beim Campusfest stand die Präsentation von Informationen zu bestimmten Themen im Vordergrund. Eine besondere Rolle nahmen dabei die Poster ein. An ihnen konnten sich Interessierte unverbindlich in das Thema einlesen und gegebenenfalls Fragen an die Aussteller stellen. Wichtig für die PG-Teilnehmer war, dass die Poster möglichst ein einheitliches Design besitzen und vor allem verständlich in das Thema IPD einführen sollen, ohne die Interessenten zu verschrecken. Die PG-Teilnehmer entschlossen sich darum für einen schlichten Aufbau, die wichtigsten Merkmale sind hierbei ein schwarzer Hintergrund, weiße Schrift und die Schlagwörter in Orange, damit sie hervorstechen. Die PG-Teilnehmer empfanden, dass vier große Poster dafür genügen sollten. Die Poster hatten folgende Themen als Schwerpunkte: Einführung, Beispiel, Praxis und Strategien.

Das Poster „IPD“ (siehe Abb. 4.5) sollte einen leichten Einstieg in die Thematik Gefangenendilemma ermöglichen. Die PG-Teilnehmer entschlossen sich daher dafür, das Gefangenendilemma möglichst leicht und verständlich in wenigen Sätzen zu beschreiben. Eine Tabelle sollte die Punktevergabe erläutern.

Das Poster „Ein kleines Spiel“ (siehe Abb. 4.6) sollte ein simples Spiel „Einer gegen Einen“ thematisieren. Hierzu wählten die PG-Teilnehmer eine Beschreibung der jeweiligen Strategie (*Random* und



Das Gefangenendilemma

PG474
WWW.IPD-DORTMUND.DE

Das **Gefangenendilemma** ist ein klassisches Problem aus der **Spieltheorie**. Es wurde zur Modellierung und Simulation in der Ökonomie, Biologie, Mathematik, Informatik und Politikwissenschaft eingesetzt. Die Basisvariante des Problems ist einfach zu beschreiben:

Zwei Verdächtige werden verhaftet und getrennt vernommen. Jeder von beiden wird vor die gleiche Wahl gestellt: zu **leugnen** oder zu **gestehen**. Falls nur ein Angeklagter gesteht und damit den anderen belastet, kommt er ohne Strafe davon und der andere wird mit 5 Jahren Haft bestraft. Leugnen beide, dann werden sie jeweils für 2 Jahre eingesperrt. Gestehen beide, so erhalten sie eine Strafe von 4 Jahren. Die Situation wird in Tabelle 1 dargestellt.

| A / B | leugnen | gestehen |
|----------|---------|----------|
| leugnen | 2/2 | 5/0 |
| gestehen | 0/5 | 4/4 |

Auf den ersten Blick erscheint ein Geständnis günstiger zu sein. Interessant wird das Spiel, wenn es **wiederholt durchgeführt** wird, das sogenannte **iterierte Gefangenendilemma** (Iterated Prisoner's Dilemma, IPD). Im Gegensatz zum einfachen Gefangenendilemma erlaubt es nun der Faktor Zeit, auf die Entscheidungen eines Konkurrenten zu einem weitaus späteren Zeitpunkt durch Bestrafung (Defektion) oder Belohnung (Kooperation) zu reagieren. Die Frage nach der besten Strategie kann nur unter Berücksichtigung der anderen Strategien beantwortet werden.

Wollen Sie mehr wissen?
WWW.IPD-DORTMUND.DE

Abbildung 4.5.: Poster „IPD“

PerKind) mit den jeweiligen Zügen, die im Beispiel gewählt wurden. Durch den beim Erstellen des Posters gewählten Zufall ging das Spiel unentschieden aus.

Ein Poster sollte zur Verdeutlichung erstellt und aufgehängt werden, damit den Interessenten gezeigt werden kann, dass das IPD auch Anwendungen im Alltag hat. So entstand das Poster „Das iterierte Gefangenendilemma in der Praxis“ (siehe Abb. 4.7). Hier wurde das Beispiel des Kalten Krieges gewählt.

Das Poster „Die Strategien“ (siehe Abb. 4.8) sollte einen kurzen Einblick in die Fülle von Strategien geben, die sich ein geneigter Leser überlegen kann, oder um Strategien selber zu gestalten.

Da die Poster im Hochschulrechenzentrum der Universität Dortmund gedruckt wurden, mussten sie eine Woche vor dem Campusfest fertig sein, da es vor einer universitätsweiten Veranstaltung zu langen Wartezeiten kommen kann.

4.6.6. Sponsoren und Preise

Um die Teilnahme am Turnier möglichst attraktiv zu machen, wurden Sponsoren und andere Geldquellen gesucht, die Preise finanzieren könnten. Schließlich ergaben sich Geldbeträge unterschiedlicher Höhe vom

- **Dekanat** des Fachbereichs für Informatik an der Universität Dortmund,
- dem Veranstaltungsservice „**PPS Music**“,
- dem „**Kurierservice Cirkel**“
- und vom Betreuer **Jörn Mehnen**.

Die Rechnung der Flyer wurde komplett von der AOK-Dortmund beglichen, so dass die Kosten für Werbung recht gering gehalten werden konnten.

Entsprechend viel Geld blieb für die Anschaffung der Preise. Es wurden die ersten fünf Plätze der Rangliste belohnt, wobei jeder Teilnehmer nur einen Preis gewinnen konnte und alle Strategien von PG-Teilnehmern sowie die Computer-Strategie „*Random*“ von der Gewinnvergabe ausgeschlossen waren. Die Preise im Einzelnen:

1. Platz: ein **Apple iPod Silver** mit 4 GB Speicher
2. Platz: ein **MP3-Player** mit 512 MB Speicher
3. Platz: ein **MP3-Player** mit 256 MB Speicher
4. Platz: ein **USB-Stick** mit 512 MB Speicher
5. Platz: ein **USB-Stick** mit 256 MB Speicher

Alle Preise wurden bei oder kurz nach der Preisverleihung am Stand der PG auf dem Campusfest abgeholt.

4.6.7. Presse

Nach erfolgreicher Durchführung des Turniers und der Verleihung der Preise auf dem Campusfest lag es nahe, Teile der Ergebnisse und Informationen über den Ablauf auch der Presse zur Verfügung zu stellen.

Für den „*Busy Beaver*“, eine vom Fachbereich Informatik selbst organisierte Zeitung, wurde ein Artikel geschrieben und eingereicht.

Der Wortlaut des Artikels ist hier zu lesen:



Wie sieht ein Spiel aus?

Am Beispiel von „Random“ und „Per kind“

WWW.IPD-DORTMUND.DE

Die Strategie „Random“ spielt eine zufällige Reihenfolge von Defektionen und Kooperationen. In diesem Beispiel die Reihe K,K,D,D,K,K,K,K,D.

Die Strategie „Per kind“ spielt immer dasselbe Muster, bestehend aus der Reihe K,K,D,K,K,D,K,K,D.

Wenn sie nun im Wettstreit miteinander spielen gilt für die Punkteverteilung folgende Tabelle:

| A/B | defektieren | kooperieren |
|-------------|-------------|-------------|
| defektieren | 1/1 | 5/0 |
| kooperieren | 0/5 | 3/3 |

Das Spiel in diesem Beispiel scheint ausgewogen zu sein, denn sowohl „Random“ als auch „Per kind“ haben dieselbe Punktzahl. Diese berechnet sich wie folgt:

| Strategie | Zug 1 | Zug 2 | Zug 3 | Zug 4 | Zug 5 | Zug 6 | Zug 7 | Zug 8 | Zug 9 |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Random | K | K | D | D | K | K | K | K | D |
| Per kind | K | K | D | K | K | D | K | K | D |
| | | | | | | | | | |
| Strategie | Punkte | Punkte | Punkte | Punkte | Punkte | Punkte | Punkte | Punkte | Punkte |
| Random | 3 | 3 | 1 | 5 | 3 | 0 | 3 | 3 | 1 |
| Per kind | 3 | 3 | 1 | 0 | 3 | 5 | 3 | 3 | 1 |

Daraus ergibt sich ein Punktestand von 22 Punkten für „Random“ und 22 Punkten für „Per kind“

Unentschieden!

Wollen Sie mehr wissen?
Fragen kostet nichts!

Abbildung 4.6.: Poster „Ein kleines Spiel“



Das iterierte Gefangenendilemma in der Praxis

Ergebnisse aus dem iterierten (wiederholten) Gefangenendilemma können auch außerhalb der **Spieltheorie** eingesetzt werden. Vor allem Situationen in **Wirtschaft und Politik** bieten Anwendungen. Typische Ziele sind das Finden einer geeigneten Strategie - abgestimmt auf die Situation und den Kontrahenten - sowie das Determinieren (Vorhersagen) des Verhaltens des Kontrahenten.

Beispiel: Kalter Krieg



Während des Kalten Krieges befanden sich die involvierten Mächte in einem **iterierten Gefangenendilemma**. Es musste immer wieder aufs neue entschieden werden ob der jeweilige Staat auf- oder abrüsten sollte.

Beiderseitiges Aufrüsten würde keinen militärischen Vorteil bringen, da beide Mächte gleich stark bleiben würden. **Beiderseitiges Abrüsten** hingegen brächte beiden Mächten keinen militärischen Nachteil, jedoch Vorteile

in anderen Bereichen, da die eingesparten Gelder in die Wohlfahrt des Landes fließen würden.




Ganz anders verhält sich der Gewinn und Verlust wenn nur **ein Kontrahent abrüstet, der andere hingegen aufrüstet**. Die aufrüstende Nation würde einen starken strategischen Vorteil erlangen.

Es handelt sich also um ein Gefangenendilemma, da den Kontrahenten nicht bewusst war wie lange der Kalte Krieg andauern würde und die beteiligten Staaten immer wieder über ihre Rüstungspolitik entscheiden mussten ist es sogar ein iteriertes Gefangenendilemma.

**Möchten Sie mehr wissen?
www.ipd-dortmund.de**

Projektgruppe 474 | Fachbereich Informatik

Abbildung 4.7.: Poster „Das iterierte Gefangenendilemma in der Praxis“



Strategien: Von Verrat und Vertrauen

Im Laufe der Zeit haben sich verschiedene Strategien entwickelt, mit denen man versuchte, so viele Punkte wie möglich zu erreichen. Viele Strategien sind untereinander sehr ähnlich, was an den begrenzten Entscheidungsmöglichkeiten des Spiels liegt. Es gibt Strategien, die sehr auf den Zufall vertrauen, andere sind sehr aggressiv, wieder andere sind sehr passiv. Da man lediglich auf die Züge des Gegners reagiert, ist es unmöglich, eine All-Round Strategie zu entwickeln, die gegen jede andere Strategie gewinnt. Dies ist eine der großen Herausforderungen des IPD, man muss immer neue Strategien entwickeln, stetig gegen neue Strategien die eigene weiterentwickeln.

In diesem Zusammenhang benutzt man verschiedene umgangssprachliche Begriffe, um zu verdeutlichen, wie die verschiedenen Reaktionen gedeutet werden können.

Ausnutzen heißt, den anderen immer zu verraten, wenn man weiß oder davon ausgeht, dass er stets kooperiert, um selber auf diese Weise die meisten Punkte zu erreichen. Meist führt dies zu einer Rache des Gegners, jedoch existieren einige Strategien, die genau dieses Schema nutzen.

Von **Rache** spricht man, wenn auf einen Verrat der einen Seite ein Verrat der anderen Seite folgt. Dies kann sich sehr leicht hochschaukeln, so dass nach wenigen Runden Verrat die einzige Möglichkeit ist, um nicht ausgenutzt zu werden, also überhaupt noch Punkte zu bekommen.

Von **Vertrauen** kann gesprochen werden, wenn über mehrere Runden von beiden Seiten mit dem anderen kooperiert wird. Dies stellt in den meisten Situationen die beste Möglichkeit dar, für beide Seiten möglichst viele Punkte zu sammeln.

Wenn nach einer langen Periode des Vertrauens von einer Seite wieder ein Verrat eingestreut wird, kann man dies als **austesten** bezeichnen. Dies geschieht meist in der Hoffnung, der andere reagiert weiterhin mit Kooperation, was als **Hoffnung** bezeichnet werden kann.

In neuerer Zeit wurden verschiedene Variationen des ursprünglichen IPD geschaffen, indem diverse neue Regeln eingeführt wurden oder bestehende abgeändert. Im folgenden werden einige dieser Neuerungen vorgestellt und auf die Auswirkungen für die Strategien näher eingegangen:

Es werden mehrere Runden gespielt. Auf diese Art und Weise wird der Zufall weitestgehend eliminiert.

Es wurde jedem ermöglicht, verschiedene Strategien einzusenden. Man beabsichtigte lediglich, eine größere Varianz der vorhandenen Strategien zu erreichen. Allerdings führte das dazu, dass eine völlig neue Strategie entwickelt wurde, die sogenannte **Master-Slave-Strategie**, auf die im weiteren noch näher eingegangen wird.

Die bekannten Strategien ähneln sich meist, was auf die eingeschränkte Aktionsmöglichkeiten der Teilnehmer zurückzuführen ist. Es existieren nur wenige grundlegende Strategien:

ALL-D: Unabhängig von den Entscheidungen des Gegners wird immer verraten (aus dem englischen *to defect*)

ALL-C: Unabhängig von den Entscheidungen des Gegners wird immer kooperiert (aus dem englischen *to cooperate*)

TTF-FOR-TAT: Es wird immer die Wahl vorgenommen, die der Gegner im letzten Zug getroffen hat. Dies ermöglicht es, sofort auf die Züge des Gegners zu reagieren.

MASTER-SLAVE: Hier arbeiten zwei Strategien zusammen. Die erste läßt sich immer von der zweiten ausnutzen. Auf diese Art und Weise erhält die zweite Strategie immer die meisten Punkte, wohin gegen die erste immer leer ausgeht.

Abbildung 4.8.: Poster „Die Strategien“

Das iterierte Gefangenendilemma

Endlich war es soweit: Auf dem Campusfest konnte nun jeder testen, wie gut er im „Verraten“ ist! Am Stand der Projektgruppe (PG) 474 in der EF 50 wurden zwei Rechner miteinander vernetzt und das Spiel per Beamer an eine Leinwand projiziert.

So konnte jeder das Spiel verfolgen, konnte genau sehen, wer mit wem kooperierte oder wer wen verriet. Trotz Absprachen zwischen den Spielern zeigte sich, dass häufig sehr viel Verrat „im Spiel“ war. Doch wieso? Und vor allem, was ist das Gefangenendilemma überhaupt? Und was heißt „iterieren“?

Das Gefangenendilemma, kurz IPD genannt, ist ein Spiel für zwei Personen, bei dem die von den Spielern verwandten Strategien, also die Art, wie sie ihre Entscheidungen treffen, sich auf viele Situationen im Leben anwenden lassen.

Das Spiel hat seinen Ursprung in einem gedachten Verhör zweier Gefangenen, bei dem beide unabhängig voneinander verhört werden. Die Gefangenen haben die Wahl, die ihnen zur Last gelegten Taten entweder zu gestehen oder zu leugnen. Allerdings wird jede Aussage in Abhängigkeit von der Aussage des anderen Mitgefangenen bewertet, wobei sich beide Personen nicht miteinander verständigen dürfen.

Das Dilemma besteht nun darin, dass beide eine immer noch erhebliche Strafe in Kauf nehmen müssen wenn beide gestehen. Wenn jedoch beide leugnen, also mit dem anderen Gefangenen zusammenarbeiten, erhalten beide eine geringere Strafe als geständen beide.

Diese ursprüngliche Idee gilt nur für jeweils eine Runde des Spiels. Um die Sache spannender zu gestalten, wird das Spiel iteriert. Das heißt, es werden mehrere Runden gespielt, wobei kein Spieler weiß, wann das Spiel genau aufhört. Es wird also eine unbekannte Anzahl an Runden gespielt.

Das Spiel ist in der iterierten Version wesentlich interessanter, da es nun möglich ist, dieselben Strategien, mit denen Entscheidungen im IPD getroffen werden, auch auf andere Situationen anzuwenden.

Ein Beispiel ist die „Organization of the Petroleum Exporting Countries (OPEC)“, gegründet am 14.10.1960 auf einer Konferenz in Bagdad. Die OPEC besteht aus mittlerweile elf Öl fördernden Ländern. Sie hat das Ziel, die Fördermengen und die Preise zu regulieren, also sicher zu stellen, dass niemand aus der Reihe tanzt und jeder genug Öl bekommt. Ein weiteres Ziel ist natürlich, dass jeder, der etwas investiert auch genug Kapital erwirtschaftet. Problematisch ist es, wenn die Ölfördermenge strikt festgelegt wird, da damit auch der Preis bestimmt wird. Je mehr gefördert wird, umso mehr Öl wird verkauft; mit der Konsequenz, dass der Preis sinkt.

Wie kann auf das OPEC-Beispiel nun das IPD angewendet werden? Jedes der Länder der OPEC stellt einen Spieler im IPD dar. Angenommen, jede Runde dauert einen Monat. Jedes Land der OPEC (d. h. jeder Spieler) kann jeden Monat aufs Neue entscheiden, ob es sich an die Absprachen hält, ob es mit den anderen kooperiert, oder ob es lieber mehr produziert und damit auch einen wesentlich höheren Gewinn erwirtschaftet als die anderen. Verstößt das Land jedoch gegen die Abmachungen, wird es aufpassen müssen, ob die anderen Länder diese Strategie nicht in den nächsten Runden ebenfalls verwenden. Das rächt sich dann in sinkenden Ölpreisen und damit langfristig in einem allgemein verringerten Gewinn.

Es zeigt sich also, dass sich das IPD auch auf Situationen, die auf den ersten Blick nichts mit Gefangenen oder einem Gefängnis zu tun haben, anwenden lässt.

Auf dem Stand konnte jeder das Problem praktisch ausprobieren. Aus Platzgründen konnten immer nur zwei Spieler gegeneinander spielen. Doch auch so haben viele Leute erste Einblicke in das IPD bekommen und – was viel wichtiger ist – allen hatten ihren Spaß dabei!



Abbildung 4.9.: Die Gewinner

Im Hintergrund lief dabei noch ein größerer Wettbewerb, bei dem über 30 Spieler teilgenommen haben. Dabei wurden von jedem Spieler ein oder zwei Strategien in Form eines Programms oder als Text (sog. Pseudo-Code) eingereicht. Das Programm hat dann in jeder Runde ermittelt, welche Entscheidung getroffen werden sollte. Nach jeder Runde wurden die Punkte verteilt. Am Ende wurden alle Punkte zusammengerechnet. Die Gewinner und alle weiteren Informationen können auf der Homepage der PG 474 www.ipd-dortmund.de nachgelesen werden. Evtl. wird Ende des WS 05/06 noch mal ein Turnier durchgeführt, bei dem dann wieder Preise gewonnen werden können. Beim vergangenen Wettbewerb konnten ein iPod, zwei mp3-Player sowie zwei USB-Sticks gewonnen werden. Zu dem neuen Wettbewerb sind natürlich alle Leser und Interessierte herzlich eingeladen, mitzumachen und zu gewinnen!

4.7. Der Stand am Campusfest

Das Campusfest selbst ist eine Art *Tag der offenen Tür*, an dem jeder, der der Universität Dortmund angehört, Informationen auf dem Campusfest zur Schau stellen kann. Da sich alle Teilnehmer der PG 474 einig darüber waren, dass es gut wäre, einen Wettbewerb mit anschließender Preisverleihung zu veranstalten, war das Campusfest die erste Wahl für ein Turnier. Die Präsentation auf dem Campusfest sollte jedoch nicht nur aus einem Wettbewerb und anschließender Preisverleihung bestehen, sondern auch noch aus einem Standspiel und Postern mit Informationen über das IPD.

Das Standspiel bestand aus drei Laptops, einer der Rechner übernahm die Aufgabe des Servers. Dieser Rechner war zuständig für die Ausgabe auf dem Beamer. Im Spiel zeigte dieser Rechner welcher Zug als letztes gemacht wurde. Am Server wurden aber auch noch alle weiteren Einstellungen für das Spiel getroffen, z. B. wer gegen wen spielt, wie viele Runden gespielt werden und ob sich der Computer am Spiel beteiligen soll.

Der Standdienst wurde so organisiert, dass zu jedem Zeitpunkt immer drei PG-Teilnehmer am Stand bereitstanden, um Fragen zu beantworten.

4.8. Fazit

Das Turnier und der damit verbundene Auftritt auf dem Campusfest konnten als Erfolg verbucht werden. Einerseits war die Wiederholung des Originalexperiments Teil der vorgegebenen Aufgaben der Projektgruppe, andererseits aber auch ein Mittel zur Datengewinnung. Durch die Strategien von über 30 Teilnehmern, die größtenteils zwei Strategien eingereicht haben, lag eine gute Basis für Analysen dieser und zukünftiger Turnierumgebungen vor. Die PG selbst war von der regen Teilnahme positiv überrascht und führt diese nicht zuletzt auf die gut organisierte und über verschiedene Medien verteilte Werbung zurück. Abgesehen vom Einsatz der Radiowerbung sind alle Möglichkeiten wie geplant durchgeführt worden.

Der Stand der PG auf dem Campusfest war ebenfalls gut organisiert. Der Ablauf konnte ohne größere Probleme durchgeführt werden. Auch hier war es positiv überraschend, dass so viele Interessierte den Weg zum Stand gefunden hatten. Bei der Preisverleihung waren alle Gewinner anwesend und konnten ihre Preise persönlich in Empfang nehmen. Einzig das Interesse der Presse, sowohl während wie auch nach dem Campusfest, hätte größer sein dürfen.

5. Ergebnisse des Campusfestturniers

Die Resonanz auf unser Campusfest-Turnier war erfreulich hoch. Insgesamt konnten 32 Teilnehmer, die Strategien eingesandt haben, gefunden werden. In drei Fällen wurde die Möglichkeit genutzt Strategien in Pseudo-Code einzusenden. Die restlichen Teilnehmer programmierten ihre Strategien in Java-Code. Sechs Mitglieder der PG nutzten außerdem die Möglichkeit mit eigenen Strategien am Turnier zu partizipieren. In acht Fällen wurde nur eine Strategie eingesandt, so dass diese Strategien jeweils geklont wurden. Hieraus resultierte eine Gesamtzahl von 65 Strategien (inklusive der von uns eingeführten Strategie *Random*).

Alle Teilnehmer kamen aus dem deutschen Raum. Ein Großteil davon (über 40 Strategien) wurde von Mitgliedern (vornehmlich Studenten) der Universität Dortmund gestellt (unter anderem auch die ersten fünf Plätze). Auch Mitglieder verschiedener informatikfremder Fachbereiche der Universität Dortmund (z. B. die Fachbereiche Wirtschaftswissenschaften oder Sport) nahmen teil. Neben 31 Männern nahm eine Frau am Wettbewerb teil. Alle Teilnehmer waren bereit, ihren Quelltext für Forschungszwecke frei zu geben.

Die Größe der erzeugten Java-Klassen lag bei minimalen 732 Byte und maximalen 7.565 Bytes. Mehr als die Hälfte der Strategien wurden am Tag des Einsendeschlusses eingesandt.

5.1. Klassifikation von Strategien

Um eine Analyse der Ergebnisse des Turniers zu ermöglichen, ist es sinnvoll die Strategien nach bestimmten Kriterien zu kategorisieren. So hat z. B. Axelrod [Axe05] für eine gute Strategie gefordert, dass diese nachgiebig sein sollte. Dies ist also ein erstes Kriterium, da auf diese Weise bei der späteren Analyse untersucht werden kann, ob die Teilnehmer sich an diese Empfehlung gehalten haben und ob solche Strategien bei unserem Turnier tatsächlich besser abgeschnitten haben.

Im Folgenden wird zwischen zwei Arten von *Attributen* unterschieden: **harten** und **weichen** Attributen. Erstere können genau beschrieben werden, für eine gegebene Strategie kann also eindeutig entschieden werden, ob sie zu einer Kategorie gehört oder nicht. Dazu werden die Kriterien genau definiert. Weiche Attribute werden nur umschrieben, da es für sie keine allgemein sinnvollen genauen Definitionen gibt. So fordert Axelrod auch, dass Strategien verständlich sein sollten. Eine gute Metrik für die Verständlichkeit von gegnerischen Strategien, welche dem intuitiven Verständlichkeitsbegriff entspricht, wurde unserem Wissen nach jedoch noch nicht entwickelt.

5.1.1. Harte Attribute

Es folgt eine Liste der harten Attribute:

- Eine Strategie ist entweder **dynamisch** oder **statisch**. Zusätzlich zum Attribut **dynamisch** kann die Strategie noch das Attribut (**bleibend**) **teilstatisch** erhalten:
 - Eine Strategie heißt **dynamisch**, wenn es zu jedem Zeitpunkt mindestens zwei verschiedene mögliche nächste Zugfolgen des Gegners gibt, welche in unterschiedlichen Wahrscheinlichkeitsverteilungen der „Antworten“ der betrachteten Strategie resultieren. (Beispiel: *TFT*)

- Eine Strategie heißt **statisch**, wenn alle möglichen unterschiedlichen Zugfolgen des Gegners nur genau eine Wahrscheinlichkeitsverteilung auf den Antworten der betrachteten Strategie zur Folge hat. (Beispiel: *PerKind*)
- Eine Strategie heißt **(bleibend) teilstatisch**, wenn es eine Zugfolge des Gegners gibt, welche die Strategie mit einer Wahrscheinlichkeit größer Null in einen Zustand bringt, ab dem sie sich immer statisch verhält. (Beispiel: Slave-Strategien defektieren immer, nachdem die Erkennung des Masters fehlgeschlagen ist).
- Eine Strategie ist entweder **deterministisch** oder **randomisiert**:
 - Eine Strategie heißt **deterministisch**, wenn sie bei jeder vom ersten Zug an gleichen Zugfolge des Gegners auch immer identisch antwortet.
 - Eine Strategie heißt **randomisiert**, wenn sie nicht deterministisch ist.
- Eine Strategie kann **periodisch** oder **(bleibend) teilperiodisch** sein:
 - Eine Strategie heißt **periodisch**, wenn es für jede Folge von Zufallszahlen, welche die Strategie erhält, eine unendliche Teilmenge $A \subseteq N$ gibt, so dass sowohl A als auch die Antwort in Zug $a \in A$ nicht von den bisherigen Zügen des Gegners abhängt. Insbesondere sind in dieser Definition auch wechselnde Periodenlängen erfasst und Phasen, in der die Strategie die bisherigen Züge des Gegners auswerten kann. Nicht erfasst ist jedoch z. B. das abwechselnde Spielen von *Pavlov* und *TFT*.
 - Eine Strategie heißt **teilperiodisch**, wenn es eine Zugfolge des Gegners gibt, welche die Strategie mit einer Wahrscheinlichkeit größer Null in einen Zustand bringt, in welchem sie sich periodisch verhält. Ab dem Erreichen des Zustandes muss es eine Fortsetzungszugfolge des Gegners geben, ab der die Strategie diesen Zustand mit einer Wahrscheinlichkeit > 0 nicht mehr verlässt. Ist die Strategie zusätzlich **bleibend teilperiodisch**, verlässt sie diesen Zustand nicht mehr.
- Eine Strategie kann **Master** oder **Slave** sein. Ist eine Strategie **Master-Strategie** dann versucht sie eine andere so genannte **Slave-Strategie** anhand ihres Verhaltens zu erkennen. Dies geschieht durch eine vorgegebene Folge von Zügen, welche von Master und Slave gespielt werden. Hat eine Master-Strategie eine Slave-Strategie erkannt, wechselt sie zur ständigen Defektion. Ausnahme ist, wenn die Slave-Strategie einen nicht vorgesehenen Zug spielt, also gar keine Slave-Strategie ist. Analog dazu wechselt eine Slave-Strategie in einen Modus der ständigen Kooperation, zumindest, bis die Master-Strategie einen Zug macht, der nicht vorausgesehen wurde. Diese Definition schließt nicht aus, dass der Master zur Erkennung z. B. einfach *TFT* spielt und dann in einer bestimmten Runde anfängt zu defektieren.

Master-Strategien können konkret daran erkannt werden, dass sie in drei Zustände aufgeteilt werden können:

1. Ständige Defektion
2. Das Spielen einer vorgefertigten Sequenz zur Erkennung
3. Eine Ausweichstrategie

Slave-Strategien müssen analog dazu ebenfalls in drei Zustände aufgeteilt werden können:

1. Ständige Kooperation
2. Das Spielen einer vorgefertigten Sequenz zur Erkennung
3. Eine Ausweichstrategie

Master-Strategien müssen immer zusammen mit „passenden“ Slave-Strategien auftauchen. Wichtig ist, dass sich Master oder Slave entweder in der Erkennungsphase befinden (nicht notwendigerweise beide), oder der Master den Slave ausnimmt. Das Spielen der vorgefertigten Erkennungssequenz (hier kann es auch mehrere geben) muss nicht zwingend im ersten Zug beginnen.

Wenn Master- und Slave-Strategien als Ausweichstrategie klassische Strategien haben, werden diese zusätzlich angegeben.

Die Erkennungssequenz sowie der Zeitpunkt, wann sie ausgespielt wird, und der Name der zugehörigen Slave-Strategie werden auch zusätzlich mit angegeben. Erkennt eine Master- oder Slave-Strategie nur, also spielt sie nicht von sich aus eine Erkennungssequenz, wird diese auch nicht angegeben.

- Eine Strategie heißt **mit Substrategie „SubStrat“**, wenn es eine Gegner-Zugfolge gibt, bei der eine positive Wahrscheinlichkeit besteht, dass die Strategie nach dieser Gegner-Zugfolge fortan immer „SubStrat“ anwendet.
- Eine Strategie heißt **freundlich**, wenn sie nie als Erste defektiert.
- Eine Strategie heißt **zurückschlagend (nach PG474)**, wenn sie nach ein oder mehreren Defektionen des Gegners in endlicher Zeit eine bestimmte, davon abhängige Anzahl von Defektionen spielt, wobei diese nicht notwendigerweise hintereinander gespielt werden müssen. Die Anzahl dieser so gespielten Defektionen muss mindestens einmal größer Null sein.
- Eine Strategie heißt **im ersten Zug kooperierend**, wenn sie immer im ersten Zug kooperiert. Eine Strategie heißt **im ersten Zug defektierend**, wenn sie immer im ersten Zug defektiert.

5.1.2. Weiche Attribute

Die weichen Attribute werden folgendermaßen umschrieben:

- Eine Strategie ist **schwach randomisiert**, wenn sie zwar Zufallszahlen benutzt, diese aber nur wenig Einfluss auf die von ihr gespielten Züge haben. Haben diese großen Einfluss, ist sie **stark randomisiert**.
- Eine Strategie ist **aggressiv**, wenn sie trotz gegnerischer Bestrafung nach eigener Defektion immer wieder (zumindest zwischendurch) Defektion spielt.
- Eine Strategie ist **versöhnlich**, wenn der Erwartungswert an Defektionen nach einer Sequenz von Defektionen des Gegners geringer als die Zahl dieser Defektionen ist.
- Eine Strategie ist **kooperativ**, wenn sie nach einer Serie von Kooperationen des Gegners ebenfalls so lange kooperiert, bis dieser irgendwann einmal defektiert.
- Eine Strategie ist **wagemutig**, wenn sie trotz gegenseitiger Kooperationen irgendwann defektiert, um das Verhalten des Gegners besser „kennen zu lernen“. Nach einer gewissen Anzahl an Bestrafungen des Gegners durch Defektionen muss sie jedoch damit aufhören, ansonsten ist sie nicht mehr wagemutig, sondern **aggressiv**.
- Eine Strategie ist **total selbstmordgefährdet**, wenn sie sehr schlecht mit Gegnern auskommt, welche für viele Strategien typische Verhaltensweisen, wie z. B. Bestrafung (Beispiel: *TFT*) oder ständige Defektion (Beispiel: *ALLD* oder Slave-Strategie) zeigen.
- Eine Strategie heißt **nachsichtig (nach Axelrod)**, wenn sie Bereitschaft zeigt nach einer Defektion des Gegenspielers erneut zu kooperieren. Nachsicht kann informell als Neigung beschrieben werden, in den Zügen nach einer Defektion des anderen zu kooperieren.

- Eine Strategie ist **verständlich (nach Axelrod)**, wenn sie für den Gegenspieler außergewöhnlich leicht zu begreifen ist. Sie macht das eigene Verhalten für den Gegner durchschaubar.
- Eine Strategie heißt **zurückschlagend (nach Axelrod)**, wenn sie nach einer „unnötigen“ Defektion der anderen Seite sofort defektiert. Was genau mit „unnötig“ gemeint ist, ist nicht präzise festgelegt.
- Eine Strategie ist **angepasst an die konkrete Spiellängenverteilung**, wenn sie die Anzahl der bisherigen Runden mitzählt und ihr Verhalten mit der Zeit ändert. Dabei müssen die in den Regeln festgelegten Mindestrundenzahlen und/oder Abbruchwahrscheinlichkeiten eine Rolle spielen. Beispielsweise könnte die Strategie erst ab Runde 1000 aggressiv werden, da es bei Mindestrundenzahlen von 100, 500 und 1000 mit anschließender Abbruchwahrscheinlichkeit von 0.1, 0.02 und 0.01 vor jeder Runde unwahrscheinlich ist, dass das Spiel zwischen Runde 900 und 1000 endet.
- Eine Strategie hat eine **Erkennung für AndereStrategie**, wenn sie im Laufe des Spieles eine Sequenz spielt, mit welcher sie herausfinden kann, ob das Verhalten ihres Gegners, zumindest in dieser Sequenz, dem von *AndereStrategie* entspricht und in diesem Falle fortan auf *AndereStrategie* angepasste Züge spielt. Diese Sequenz muss nicht fest eingegeben sein, es reicht auch, wenn die Strategie sich merkt, ob der Gegner *AndereStrategie* ist und, sofern dies erfüllt ist, nach irgendwelchen anderen Kriterien später dazu übergeht, die angepassten Züge zu spielen.
- Eine Strategie ist ein **Kooperativer Slave**, wenn sie eine Slave-Strategie ist und Gegnern, welche nicht ihr Master sind, mit eher kooperativen Ausweichstrategien gegenüber tritt. Im Gegensatz dazu kann sie ein **Unkooperativer Slave** sein; dabei versucht sie Gegnern, welche nicht ihr Master sind, durch häufige Defektion zu schädigen.
- Eine Strategie ist **evolutionär optimiert**, wenn sie Parameter enthält, welche vom Programmierer offen gelassen wurden. Diese Parameter wurden dann mit heuristischen Suchverfahren bestimmt. Die Zielfunktion bildet dabei eine Variation der Turnierumgebung, wobei bestimmte Gegner vorgegeben waren. Ein dabei gefundener guter Parametersatz wird dann für die Turnierstrategie verwendet.

5.1.3. Standardstrategien

Einige der genannten Attribute werden durch (eine Liste von) Standardstrategien parametrisiert. Aus Gründen des Aufwands werden bei solchen Attributen jedoch nur die folgenden Standardstrategien betrachtet:

- *ALLC* (Always cooperate)
- *ALLD* (Always defect)
- *TFT* (Tit-for-Tat)
- *TfTT* (Tit-for-two-Tats)
- *GRADUAL*
- *Majority*

5.1.4. Strategien von PG-Teilnehmern

In diesem Abschnitt werden die von den PG-Teilnehmern zum Campusfestturnier selbst entwickelten Strategien beschrieben. Es wurde sich zu dieser Vorgehensweise entschieden, da für eine umfassende Gruppenarbeit aufgrund der geringen Zeitspanne bis zum Einsendeschluss nicht mehr genügend Zeit war. Außerdem wurden stark unterschiedliche Ansätze der einzelnen Teilnehmer als Anregungen für spätere Entwürfe erhofft, welche bei Gruppenarbeit möglicherweise als Konsequenz der Konsensfindung nicht realisiert worden wären.

5.1.4.1. „Throat“

Autor: Bastian Baranski

Platz: 46

Harte Attribute: bleibend teilstatisch, Master mit Slave „Deep“, zehn Züge Erkennungssequenz am Anfang, mit Substrategien *ALLD* und *ALLC*, zurückschlagend

Weiche Attribute: evolutionär optimiert

Nicht analysierbare Attribute: periodisch, mit anderen Substrategien, aggressiv, versöhnlich, nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod), Erkennung für andere Strategien

Als erstes muss erwähnt werden, dass der Autor dieser Strategie keine Erfahrung mit evolutionären Optimierungsverfahren hat. Die Entwicklung dieser Strategie war für den Autor der erste Berührungspunkt mit dieser Art von Optimierungsstrategien. Fehler bzw. Ungenauigkeiten in der Implementierung der evolutionären Umgebung sind die Folge dieser fehlenden Erfahrung.

Die Idee hinter der Strategie ist, die vorhandenen Vorteile einer normalen Master-Slave Strategie mit den Vorteilen einer evolutionären Optimierung zu verknüpfen.

Bei der evolutionären Optimierung hat der Autor eine „Standardkodierung“ für das iterierte Gefangenendilemma genommen. Die Strategie sollte anhand der letzten zehn gespielten Zugkombinationen den nächsten Zug berechnen. Für jede mögliche Zugkombinationen wurde ein zu spielender Folgezug gespeichert („0“ für Kooperation und „1“ für Defektion). Dieser Bit-Vektor repräsentierte die DNA des genetischen Algorithmus.

Jede Strategie aus einer Population musste in einem Turnier gegen die Strategien der Testumgebung antreten. Abhängig von dem Erfolg gegen die Strategien aus der Testumgebung wurden die Individuen bzw. die Strategien reproduziert. Die genetischen Informationen der Individuen wurden mit verschiedenen genetischen Operatoren modifiziert.

Zur Verbesserung der Chancen im Wettbewerb wurde Throat weiterhin auf klassische Art und Weise zu einer Master-Slave-Strategie ausgebaut. Der Slave trägt den Namen „Deep“. Der Master fängt mit der Erkennungssequenz an und spielt im Fall, dass der Gegner einen für den Slave oder Master nicht vorgesehenen Zug macht, mit der oben beschriebenen Strategie weiter. Ansonsten wechselt er nach Beendigung der Erkennungssequenz in einen Modus ständiger Defektion. Zusätzlich ist der Master noch in der Lage, durch eben diese Sequenz sich selbst zu erkennen und wechselt in diesem Fall nach der Erkennungssequenz in einen Zustand ständiger Kooperation. Die Erkennungssequenzen wurden ebenfalls evolutionär optimiert.

5.1.4.2. „Deep“

Autor: Bastian Baranski

Platz: 64

Harte Attribute: bleibend teilstatisch, deterministisch, Slave mit Master „Throat“, zehn Züge Erkennungssequenz am Anfang

Weiche Attribute: aggressiv, total selbstmordgefährdet, zurückschlagend (nach Axelrod), unkooperativer Slave

Nicht analysierbare Attribute: keine

Die Strategie ist ein klassischer Slave als Gegenstück zu „Throat“. Im Falle der Nicht-Erkennung des Masters wird immer defektiert.

5.1.4.3. „Green Shark“

Autor: Jens Schmidt

Platz: 43

Harte Attribute: dynamisch, deterministisch, Master (zugehöriger Slave: „Tasty Fish“, Erkennungssequenz am Anfang: CCDCD), mit Substrategie *GRADUAL*, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: kooperativ, wagemutig, nachsichtig

Nicht analysierbare Attribute: versöhnlich

Diese Strategie ist eine Master-Strategie mit dem Slave „Tasty Fish“. „Tasty Fish“ wird über die oben angegebene Erkennungssequenz erkannt, in der „Tasty Fish“ immer kooperiert. *ALLC* würde also auch als Slave erkannt werden. Die Substrategie *GRADUAL* wird gespielt, wenn kein Slave erkannt wurde oder aber ein erkannter Slave mehr als einmal defektiert. *GRADUAL* wurde dabei etwas optimiert, indem Parameter ausgetauscht wurden.

5.1.4.4. „Tasty Fish“

Autor: Jens Schmidt

Platz: 59

Harte Attribute: bleibend teilstatisch, randomisiert, bleibend teilperiodisch, Slave (zugehöriger Master: Green Shark, Erkennungssequenz am Anfang), mit Substrategie *ALLD*, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: schwach randomisiert, aggressiv, verständlich, unkooperativer Slave

Nicht analysierbare Attribute: keine

Diese Strategie ist eine Slave-Strategie mit dem Master Green Shark. Tasty Fish kooperiert die ersten sechs Züge und suggeriert so dem Master uneingeschränkte Kooperation. Bei Mastererkennung spielt Tasty Fish *ALLC*, sonst mit einer sehr hohen Wahrscheinlichkeit *ALLD*.

5.1.4.5. „Major Eisberg“

Autor: Rüdiger Ehlers

Platz: 45

Harte Attribute: bleibend teilstatisch, randomisiert, Master mit Slave „Minor Eisberg“, elf Züge Erkennungssequenz am Anfang, im ersten Zug kooperierend, mit Substrategien *ALLD* und *ALLC*, zurückschlagend (nach PG474, experimentell ermittelt),

Weiche Attribute: schwach randomisiert, evolutionär optimiert

Nicht analysierbare Attribute: periodisch, mit anderen Substrategien, aggressiv, versöhnlich, nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod), Erkennung für andere Strategien

Idee hinter dieser Strategie ist es, gefühlsgelitetes intuitives Verhalten, wie es bei realen höheren Lebewesen zu finden ist, zu simulieren. Allerdings bleibt es zunächst völlig offen, um welche Gefühle es sich handeln soll (beispielsweise Zorn, Freundlichkeit, etc.). Dies wurde erst durch die evolutionäre Generierung der Parameter entschieden.

Die Stärke eines Gefühls wird durch einen Fließkommawert repräsentiert, welcher sich in jeder Runde einmal ändert. Die Initialisierungswerte sind vorgegebene Parameter. Zu jedem Gefühl ist vorgegeben, welcher Wert für welche Kombination aus den eigenen letzten beiden Zügen und den letzten beiden Zügen des Gegners hinzu addiert werden soll. Da es biologisch motiviert ist, dass die Stärke eines Gefühls nicht beliebig hoch sein kann, wird noch nach der Aktualisierung des Wertes in jeder Runde eine sogenannte Sigma-Funktion nachgeschaltet, welche dafür sorgt, dass mit der Zeit bei fehlender Stimulation (d. h. bei Hinzuaddierung von Nullwerten) der Wert des Gefühls gegen 0.5 konvergiert und sich immer stets zwischen 0 und 1 befindet.

Es gibt insgesamt vier Gefühle. Um nun die Entscheidung über Kooperation und Defektion zu treffen, werden die Werte der Gefühle mit weiteren Parametern multipliziert und addiert. Dazu kommt noch eine ebenfalls durch einen Parameter gewichtete Zufallszahl. Ist dieser Gesamtwert nun unter einem bestimmten Schwellwert, wird kooperiert, ansonsten defektiert.

Die Aktualisierungstabelle hat die Größe $2 \times 2 \times 2$, das sind bereits 64 Parameter (d. h. 16 pro Gefühl). Dazu kommen noch die Initialisierungswerte (vier Stück), die Gewichtungen der Gefühle und der Zufallszahl (insgesamt fünf Parameter) und der Schwellwert. Die Sigma-Funktion ist ebenfalls für jedes Gefühl einzeln parametrisiert, wodurch sich „Stimmungen“ je nach Wert, schneller oder langsamer „normalisieren“. Insgesamt gibt es also 78 Parameter, welche vorerst unbestimmt waren. Für diese wurde dann in einem pseudo-evolutionären Ansatz (Simulated Annealing) eine gute Belegung gesucht. Dazu wurden Turniere mit Standardstrategien ausgetragen und bessere Parametersätze bevorzugt weiterverfolgt. Durch das geringe „Gedächtnis“ der Strategie trat nur eine geringe Spezialisierung auf die konkreten Strategien auf. Die in ihnen vorhandenen typischen Verhaltensweisen konnten so trainiert werden, so dass die Strategie für höhere Lebewesen typische Verhaltensweisen zeigt, welche sich wiederum gut im IPD schlagen.

Zur Verbesserung der Chancen im Wettbewerb wurde Major Eisberg weiterhin auf klassische Art und Weise zu einer Master-Slave-Strategie ausgebaut. Der Slave trägt den Namen „Minor Eisberg“. Der Master fängt mit der Erkennungssequenz an und spielt im Fall, dass der Gegner einen für den Slave oder Master nicht vorgesehenen Zug macht, mit der oben beschriebenen Strategie weiter. Ansonsten wechselt er nach Beendigung der Erkennungssequenz in einen Modus ständiger Defektion. Zusätzlich ist der Master noch in der Lage, durch eben diese Sequenz sich selbst zu erkennen und wechselt in diesem Fall nach der Erkennungssequenz in einen Zustand ständiger Kooperation. Die Erkennungssequenzen wurden nicht evolutionär optimiert.

5.1.4.6. „Minor Eisberg“

Autor: Rüdiger Ehlers

Platz: 65

Harte Attribute: bleibend teilstatisch, deterministisch, Slave mit Master „Major Eisberg“, elf Züge Erkennungssequenz am Anfang, im ersten Zug kooperierend

Weiche Attribute: aggressiv, total selbstmordgefährdet, zurückschlagend (nach Axelrod), unkooperativer Slave

Nicht analysierbare Attribute: keine

Die Strategie ist ein klassischer Slave als Gegenstück zu „Major Eisberg“. Im Falle der Nicht-Erkennung des Masters wird immer defektiert.

5.1.4.7. „Simon“

Autor: Simon Steeg

Platz: 18

Harte Attribute: dynamisch, deterministisch, freundlich, teilperiodisch, im ersten Zug kooperierend, zurückschlagend (nach PG474)

Weiche Attribute: nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie ähnelt einer *GRADUAL* Strategie, betrachtet jedoch nur die letzten fünf Züge des Gegners. Die Strategie zählt die Anzahl der Defektionen in den letzten fünf Zügen und errechnet daraus die Anzahl der eigenen folgenden Defektionen. Diese werden errechnet, indem einfach die Anzahl der Defektionen des Gegners (in den letzten fünf Runden) mit sich selbst multipliziert werden. Dann spielt die Strategie diese Defektionen gefolgt von zwei Kooperationen. Nachdem dieses Muster gespielt wurde, erfolgt die Berechnung erneut aus den letzten fünf Zügen des Gegners.

5.1.4.8. „Simon2“

Autor: Simon Steeg

Platz: 13

Harte Attribute: dynamisch, deterministisch, freundlich, teilperiodisch, im ersten Zug kooperierend, zurückschlagend (nach PG474)

Weiche Attribute: nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie ist fast identisch mit „Simon“. Der Unterschied liegt darin, dass diese Strategie nur eine Kooperation nach der Defektionsphase spielt.

5.1.4.9. „He-Man“

Autor: Thusinthan Kajendran

Platz: 14

Harte Attribute: teilstatisch, deterministisch, Master mit Slave „Die Macht von Grayskull“, teilperiodisch, zurückschlagend (nach PG474), mit Substrategie *TFT*, im ersten Zug Kooperierend

Weiche Attribute: nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie hat drei Zustände: Mastermode, Equalmode und Normalmode. Sie spielt zuerst eine eindeutige Sequenz von Zügen, in diesem Fall CDCDCCC. Dabei beobachtet die Strategie, welche Züge der Gegner macht. Wenn die eben genannte Sequenz erkannt wird, bedeutet das für „He-Man“, dass dieser gegen sich selbst spielt und somit wechselt die Strategie in den Equalmode. In diesem Zustand spielt die Strategie nur Kooperation, um damit keine Punkte zu verlieren. Die zweite Möglichkeit ist, dass die Strategie die Sequenz des zuhörigen Slaves (CCCCDC) erkennt. In diesem Fall wechselt sie in den Mastermode und defektiert dann fortan, denn der Slave spielt immer Kooperation, wenn dieser seinerseits die Master-Sequenz erkannt hat. Die dritte Möglichkeit ist, dass keine der beiden Sequenzen erkannt wird, dann wechselt „He-Man“ in den Normalmode und spielt fortan die klassische Strategie *TFT*.

5.1.4.10. „Die Macht von Grayskull“

Autor: Thusinthan Kajendran

Platz: 58

Harte Attribute: teilstatisch, deterministisch, Slave mit Master „He-Man“, teilperiodisch, mit Substrategie *ALLD*, im ersten Zug Kooperierend

Weiche Attribute: wagemutig, total selbstmordgefährdet, nachsichtig, verständlich, kooperierender Slave

Nicht analysierbare Attribute: keine

Diese Strategie ist der zu „He-Man“ gehörende Slave. Sie hat, genau wie die Master-Strategie, drei Zustände in welche sie geraten kann: Slavemode, Equalmode und Normalmode. Wie bereits bei der Beschreibung des Masters erwähnt, spielt die Slave-Strategie die Erkennungssequenz CCCCCDC. Wird dieser Code erkannt, bedeutet dies für die Strategie, dass sie gegen sich selbst spielt und sie wechselt somit in den Equalmode (spielt Kooperation). Wenn jedoch der Mastercode erkannt wurde, spielt sie ständig Kooperation. Falls kein Code erkannt wurde, spielt sie *ALLD*, um damit möglichst wenig Punkte an den Gegner zu vergeben.

5.1.4.11. „Young Capone“

Autor: Marc Zimmermann

Platz: 42

Harte Attribute: dynamisch, deterministisch, Master (zum Slave „Betonschuh“), im ersten Zug kooperierend

Weiche Attribute: nachsichtig, zurückschlagend (nach Axelrod), Erkennung für: „Betonschuh“, *ALLC*, *ALLD*, Per Kind, Per Nasty, Go By Majority, Spite, *TFT* und *TfTT*

Nicht analysierbare Attribute: keine

„Young Capone“ ist die Master-Strategie zu „Betonschuh“. Beide spielen eine Erkennungssequenz und „Young Capone“ bekommt im Falle eines Spieles zwischen den beiden auf eine Defektion immer Kooperation als Antwort. Zudem versucht die Strategie, möglichst viele gängige Strategien zu erkennen und gegen diese möglichst optimal zu spielen. Kann „Young Capone“ keinen Gegner klar identifizieren spielt er *TFT*, kooperiert aber in jeder elften Runde, um sich versöhnlich zu zeigen.

5.1.4.12. „Betonschuh“

Autor: Marc Zimmermann

Platz: 60

Harte Attribute: dynamisch, deterministisch, Slave (zum Master „Young Capone“), im ersten Zug defektierend

Weiche Attribute: aggressiv, zurückschlagend (nach Axelrod), verständlich, unkooperativer Slave, Erkennung für: Young Capone, *ALLC*, *ALLD*, Per Kind, Per Nasty, Spite, *TFT* und *TfTT*

Nicht analysierbare Attribute: keine

„Betonschuh“ ist der Slave zum Master „Young Capone“. Die Strategie ist im Grunde ähnlich aufgebaut und versucht andere Strategien zu erkennen. Sie spielt *ALLC* gegen den Master, versucht aus Strategien, die sich ausnutzen lassen (z. B. *ALLC*, *TfTT* u. a.) Punkte zu schlagen und spielt gegen alle anderen (z. B. *TFT*) *ALLD*, um ihnen möglichst geringen Punktgewinn zu erlauben.

5.1.5. Strategien von Teilnehmern außerhalb der PG

Es folgen Beschreibungen der eingesendeten Strategien von Teilnehmern außerhalb der PG. Es wurde versucht, diese so gut wie möglich zu analysieren. Aufgrund der Komplexität einiger Einsendungen war dies jedoch leider nicht immer mit vertretbarem Aufwand möglich. In einem solchen Fall wird entweder konkret darauf hingewiesen oder bei wenigen Attributen unter „Nicht analysierbare Attribute“ angegeben, welche nicht untersucht wurden.

5.1.5.1. „Meister“

Platz: 1

Autor: Christian Gentges

Harte Attribute: bleibend teilstatisch, deterministisch, bleibend teilperiodisch, Master (mit Slave: „Sklave“, Erkennungssequenz ab Zug sieben: CDCCDC), mit Substrategie *TFT*, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: kooperativ, verständlich, zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie spielt in den ersten sechs Zügen *TFT*. In den sechs darauf folgenden Zügen wird der Code CDCCDC als Erkennungssequenz gespielt, falls noch nicht verifiziert werden konnte, ob gegen die Strategie „Sklave“ gespielt wird. Falls ab Zug 13 der Slave erkannt worden ist, wird danach immer defektiert, andernfalls immer *TFT* gespielt.

5.1.5.2. „Netzer“

Platz: 2

Autor: Jan Schultze

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.3. „MN1“

Autor: Michael Nöthe

Platz: 3

Harte Attribute: bleibend teilstatisch, deterministisch, bleibend teilperiodisch, Master (mit Slave: „MN2“, Erkennungssequenz ab Zug eins), mit Substrategie *TFT*, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: kooperativ, verständlich, zurückschlagend

Nicht analysierbare Attribute: keine

Diese Master-Strategie hält alle bisher gespielten Züge in einer Liste fest. Generell wird *TFT* gespielt, allerdings mit folgender Ausnahme: Im Zug 17 werden die in den Zügen 1-15 getroffenen Entscheidungen geprüft, um den Slave „MN2“ erkennen zu können. Wurde der Slave erkannt, wird danach immer Defektion gespielt.

5.1.5.4. „TitForTwoTats“

Autor: Christian Bredemeier

Platz: 4

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, verständlich, zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie entspricht der Standardstrategie *TFT*, der Autor hat sie aber unter dem Namen „Tit-for-Two-Tats“ zwecks Verwirrung eingereicht.

5.1.5.5. „Nils-TFT“

Autor: Nils Kremeskötter

Platz: 5 und 8

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, zurückschlagend (nach Axelrod), verständlich

Nicht analysierbare Attribute: keine

Diese Strategie ist ebenfalls die Standardstrategie *TFT*.

5.1.5.6. „S-TFT“

Autor: Lars Schluecker

Platz: 6

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, verständlich, zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie ist ebenfalls die Standardstrategie *TFT*.

5.1.5.7. „Sebrassi IPD II“

Platz: 7

Autor: Christian Groß

Harte Attribute: dynamisch, randomisiert, mit Substrategie *TFT*, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: schwach randomisiert, kooperativ, verständlich, zurückschlagend

Nicht analysierbare Attribute: keine

Diese Strategie ist eine *TFT*-Variante. Es wird allerdings vor dem Spiel eine Zufallszahl im Bereich von 0 bis 100 gezogen und dann in genau dieser Runde eine Kooperation gespielt.

5.1.5.8. „Späte Rache“

Autor: Michael Hein

Platz: 9

Harte Attribute: dynamisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie kooperiert genau solange, bis der Gegner das erste Mal defektiert hat. Dann kommt sie in einen Zustand, in dem dreimal defektiert wird, worauf die Strategie wieder in ihren Startzustand übergeht. Dort startet sie jedoch mit Defektion. Bei einer Kooperation des Gegners direkt nach dieser Defektion nimmt sie dann die Kooperation wieder auf.

5.1.5.9. „THModGradual1“

Autor: Thorsten Hermes

Platz: 10

Harte Attribute: (bleibend) teilstatisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich ähnlich der klassischen *GRADUAL* Strategie. Der Unterschied liegt darin, dass die Strategie die Anzahl der Defektionen und die Anzahl der Kooperationen des Gegners zählt. Ab Runde 25 überprüft die Strategie in jeder Runde, ob die Anzahl der Defektionen des Gegners doppelt so groß wie die Anzahl der Kooperationen des Gegners ist. Ist dies der Fall, defektiert die Strategie bis zum Ende.

5.1.5.10. „THModGradual2“

Autor: Thorsten Hermes

Platz: 11

Harte Attribute: (bleibend) teilstatisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod), angepasst an die konkrete Spiellängenverteilung

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich wie „THModGradual1“, hat jedoch den Zusatz, dass sie ab Runde 1021 nur noch defektiert, wenn der Gegner im gesamten Spielverlauf bisher einmal defektiert hat.

5.1.5.11. „TitForTat Plus1“

Autor: Johanna Roskosch

Platz: 12

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, versöhnlich, zurückschlagend (nach Axelrod), verständlich

Nicht analysierbare Attribute: keine

„TitForTat Plus1“ funktioniert genauso wie „LateTfT“, jedoch schaltet diese Strategie schon nach zwei Defektionen in den *TFT*-Modus.

5.1.5.12. „Zweiundvierzig“

Autor: Matthias Miemczyk

Platz: 15

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.13. „SchwallisFaku“

Autor: Frank Schwarze

Platz: 16

Harte Attribute: (bleibend) teilstatisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich ähnlich zu der klassischen *GRADUAL* Strategie. Allerdings wird die Anzahl der Defektionen über eine andere Formel berechnet. Die Formel zur Berechnung lautet: $Defektionen = n \cdot \frac{(n+1)}{2}$ (wobei n der Anzahl der Defektionen des Gegners entspricht). Die Strategie hat noch den Zusatz, dass sie selbst nur defektiert, nachdem der Gegner insgesamt 40 mal defektiert hat.

5.1.5.14. „FJ Last-Twen Tit-for-tat“ with „TfT-Recognition“

Autor: Felix Jungermann

Platz: 17

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.15. „FJ Last-Twen Tit-for-tat“

Autor: Felix Jungermann

Platz: 19

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.16. „S-TFTT“

Autor: Lars Schluecker

Platz: 20

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, versöhnlich, zurückschlagend (nach Axelrod), verständlich

Nicht analysierbare Attribute: keine

Diese Strategie entspricht der Standardstrategie *TfTT*. Im Vergleich zu *TFT* wartet diese Strategie zwei Züge des Gegners ab und falls der der Gegner zweimal in Folge defektiert hat, defektiert diese Strategie auch.

5.1.5.17. „SchwallisGradual“

Autor: Frank Schwarze

Platz: 21

Harte Attribute: dynamisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich so wie die klassische *GRADUAL* Strategie. Der einzige Unterschied ist, dass SchwallisGradual auch im zweiten Zug noch kooperiert, unabhängig vom Verhalten des Gegners.

5.1.5.18. „Sebrassi IPD I“

Platz: 22

Autor: Christian Groß

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: kooperativ, zurückschlagend

Nicht analysierbare Attribute: keine

Diese Strategie zählt die Anzahl der Defektionen des Gegners. Neuere Defektionen werden immer schwerer gewichtet. Die Summe dieser Gewichtungen ist ausschlaggebend für eine Defektion oder Kooperation. Bei ständiger Kooperation wird die Summe pro Kooperation um eins gesenkt.

5.1.5.19. „Verzögerte Bestrafung“

Autor: Michael Hein

Platz: 23

Harte Attribute: dynamisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod), zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie kooperiert solange, bis ihr Gegner das erste Mal defektiert hat. Dann kommt sie in einen Zustand, in dem nur noch defektiert wird. Nach jeweils zehn Runden geht die Strategie wieder in ihren Startzustand über. Sie startet dann jedoch mit Defektion. Bei einer Kooperation des Gegners direkt nach dieser Defektion nimmt sie dann die Kooperation wieder auf.

5.1.5.20. „LateTFT“

Autor: Johanna Roskosch

Platz: 24

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474),

Weiche Attribute: kooperativ, versöhnlich, zurückschlagend (nach Axelrod), verständlich

Nicht analysierbare Attribute: keine

„LateTFT“ ist eine modifizierte *TFT*-Strategie. Die Spielweise ändert sich insofern, dass solange Kooperation gespielt wird bis der Gegner achtmal defektiert, danach wird *TFT* gespielt.

5.1.5.21. „Boris Konrad 1“

Autor: Boris Konrad

Platz: 25

Harte Attribute: dynamisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich ähnlich der klassischen *GRADUAL* Strategie, allerdings wird die Anzahl der Defektionen über eine andere Formel berechnet. Die Formel zur Berechnung lautet: *Defektionen* = $n \cdot \frac{(n+1)}{2}$ (wobei n der Anzahl der Defektionen des Gegners entspricht).

5.1.5.22. „Boris Konrad 2“

Autor: Boris Konrad

Platz: 26

Harte Attribute: dynamisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend, teilperiodisch

Weiche Attribute: nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie verhält sich ähnlich der klassischen *GRADUAL* Strategie, allerdings wird die Anzahl der Defektionen über eine andere Formel berechnet. Die Formel zur Berechnung lautet: *Defektionen* = $n \cdot n$ (wobei n der Anzahl der Defektionen des Gegners entspricht).

5.1.5.23. „adamone“

Autor: Adam Vincze

Platz: 27

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.24. „TBattle Version 4“**Autor:** Tim Mueller**Platz:** 28 und 29**Harte Attribute:** nicht analysiert**Weiche Attribute:** nicht analysiert**Nicht analysierbare Attribute:** nicht analysiert**5.1.5.25. „Weighted Random“****Autor:** Marco Bakera**Platz:** 30 und 31**Harte Attribute:** dynamisch, randomisiert, zurückschlagend (nach PG474), im ersten Zug kooperierend**Weiche Attribute:** stark randomisiert, kooperativ, zurückschlagend**Nicht analysierbare Attribute:** keine

„Weighted Random“ zählt die aktuelle Zugnummer und die Anzahl der Defektionen des Gegners mit. Es wird dann mit der Wahrscheinlichkeit $\frac{\text{Defektionen}}{\text{Zugnummer}}$ defektiert, ansonsten kooperiert.

5.1.5.26. „Mugling Through“**Autor:** Ralf Bochon**Platz:** 32 und 33**Harte Attribute:** dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend**Weiche Attribute:** kooperativ, zurückschlagend (nach Axelrod), Erkennung für *ALLD* und zwei periodische Muster**Nicht analysierbare Attribute:** keine

Diese Strategie hält die letzten fünf gespielten Züge des Gegners fest. Wenn bestimmte Muster in diesen fünf letzten Zügen erkannt worden sind, wird in den meisten Fällen defektiert. Die erkannten Muster sind CDCCD, DCDDC und DDDDD. Insbesondere wird also auch *ALLD* erkannt. Bei den ersten beiden (periodischen) Mustern wird nur dann kooperiert, wenn der logische nächste Zug des Gegners auch Kooperation ist. Falls der Gegner doppelt so viele Defektionen gespielt hat wie die eigene Strategie, wird eine Rache phase von Defektionen gespielt. Diese wird nur dann verlassen, falls die eigene Strategie doppelt so oft defektierte wie ihr Kontrahent. Trifft weder ein Muster noch die Rache phase zu, wird kooperiert.

5.1.5.27. „evaWobeGa“

Autor: Nils Alberti

Platz: 34 und 35

Harte Attribute: dynamisch, randomisiert, zurückschlagend (nach PG474, experimentell ermittelt), im ersten Zug kooperierend

Weiche Attribute: stark randomisiert, wagemutig, nachsichtig (nach Axelrod, experimentell ermittelt), evolutionär optimiert

Nicht analysierbare Attribute: periodisch, mit Substrategie, aggressiv, versöhnlich, nachsichtig (nach Axelrod), angepasst an die konkrete Spiellängenverteilung, Erkennung für andere Strategien

„EvaWoBeGa“ hat 56 reellwertige Parameter, für die eine gute Belegung offensichtlich durch einen evolutionären Prozess ermittelt wurde. Sie zählt das Auftreten jeder möglichen Zugkombination mit (d. h. Kooperieren/Kooperieren, Kooperieren/Defektieren, Defektieren/Kooperieren und Defektieren/Defektieren). Zusätzlich benutzt sie zur Entscheidung die eigenen letzten Züge und die beiden letzten Züge ihres Gegners. Begonnen wird grundsätzlich mit Kooperation.

Bei der eigentlichen Entscheidungsfindung betrachtet evaWobeGa die bisherige durchschnittliche Anzahl an Defektionen des Gegners und die bisherigen Zugpaarungen. Diese bilden zusammen mit Zufallsentscheidungen einen verschachtelten if-Block, in dem einige Vergleiche zwischen den eben genannten Variablen gemacht werden. Dabei werden alle vorkommenden Schwellwerte für die Entscheidung durch einen der reellwertigen Parameter repräsentiert. Für jede Kombination aus den eigenen letzten beiden Zügen und dem letzten Zug des Gegners gibt es hierbei einen eigenen Parametersatz. Obwohl ein solcher if-Block relativ kurz ist, ist die Idee hinter der Struktur nicht ohne Weiteres ersichtlich.

Da durch die große Anzahl an Parametern das Verhalten nur schwer durchschaubar ist, konnten leider nicht alle Attribute analysiert werden.

5.1.5.28. „StrategyByBas“

Autor: Bastian Reining

Platz: 36 und 37

Harte Attribute: dynamisch, randomisiert, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: stark randomisiert, versöhnlich, kooperativ, nachsichtig (nach Axelrod)

Nicht analysierbare Attribute: keine

„StrategyByBas“ ist eine sehr kooperative Strategie. Sie entspricht *TFT*, nur dass hier im Falle einer Defektion des Gegners lediglich mit einer Wahrscheinlichkeit von $\frac{2}{5} + \epsilon$ (wobei ϵ dem Rundungsfehler durch Verwendung des `double`-Datentyps von Java bei Zahlen in $[0...1]$ entspricht) zurückgeschlagen wird.

5.1.5.29. „Der Zorn Gottes“

Autor: David Schulz

Platz: 38 und 39

Harte Attribute: (bleibend) teilstatisch, freundlich, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute:

Nicht analysierbare Attribute: keine

Diese Strategie kooperiert solange, bis der Gegner insgesamt 23 Defektionen gespielt hat. Hat der Gegner diese 23 Defektionen erreicht, spielt die Strategie nur noch Defektion bis zum Ende vom Spiel.

5.1.5.30. „Sauer2“

Autor: Ulf Preising

Platz: 40

Harte Attribute: dynamisch, deterministisch, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: nachsichtig (nach Axelrod), angepasst an die konkrete Spiellängenverteilung

Nicht analysierbare Attribute: keine

Diese Strategie spielt die ersten 15 Züge immer Kooperation. Spielt der Gegner nach dieser Sequenz eine Defektion, defektiert „Sauer2“ bis zum Ende des Spiels. Es gibt die Ausnahme, dass jeweils in den Zügen 100, 500 und 1000 das Gedächtnis gelöscht wird. Danach zeigt sich „Sauer2“ nachsichtig und im Falle einer Kooperation des Gegners kooperiert die Strategie wieder. Bei einer erneuten Defektion des Gegners wird wieder auf dauerhafte Defektion geschaltet.

5.1.5.31. „Sauer1“

Autor: Ulf Preising

Platz: 41

Harte Attribute: (bleibend) teilstatisch, deterministisch, freundlich, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute:

Nicht analysierbare Attribute: keine

Diese Strategie spielt die ersten 20 Züge immer Kooperation. Spielt der Gegner nach dieser Sequenz eine Defektion, defektiert „Sauer1“ bis zum Ende des Spiels.

5.1.5.32. „Raskolnikof“

Autor: Heiko Röglin

Platz: 44

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: noch nicht analysiert

5.1.5.33. „adamtwo“

Autor: Adam Vincze

Platz: 47

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.34. „MN2“

Autor: Michael Nöthe

Platz: 48

Harte Attribute: statisch, deterministisch, bleibend teilperiodisch, Slave (mit Master: „MN1“, Erkennungssequenz ab Zug eins: DCCDDDCCCCDDDDDD), mit Substrategie *ALLC*, im ersten Zug defektierend

Weiche Attribute: versöhnlich, wagemutig, total selbstmordgefährdet, nachsichtig, verständlich, kooperierender Slave

Nicht analysierbare Attribute: keine

Diese Slave-Strategie spielt die ersten 15 Züge die Sequenz DCCDDDCCCCDDDDDD. Danach wird *ALLC* gespielt.

5.1.5.35. „Dirk Alpha“

Autor: Dirk Blotzheim

Platz: 49

Harte Attribute: dynamisch, deterministisch, periodisch, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: aggressiv, zurückschlagend (nach Axelrod)

Nicht analysierbare Attribute: keine

Diese Strategie erreichte uns als Pseudo-Code, welcher folgendermaßen lautet:

Meine Strategie soll sich verhalten wie die als *TFT* beschriebene. Ich weiche jedoch davon ab, indem bei jedem zehnten Fall defektiert anstelle von kooperiert wird.

Es gab nun zwei Möglichkeiten, diese Anweisungen zu interpretieren. „Dirk Alpha“ ist die erste Variante. Sie entspricht genau *TFT*, nur dass die bisherige Anzahl an Runden im Spiel gezählt wird und unabhängig von dem Zug des Gegners bei einer glatt durch zehn teilbaren Zugnummer defektiert wird.

5.1.5.36. „Dirk Beta“**Autor:** Dirk Blotzheim**Platz:** 50**Harte Attribute:** dynamisch, deterministisch, zurückschlagend (nach PG474), im ersten Zug kooperierend**Weiche Attribute:** aggressiv, zurückschlagend (nach Axelrod)**Nicht analysierbare Attribute:** keine

Dies ist die zweite Möglichkeit den Pseudo-Code von Dirk Blotzheim zu interpretieren. Beide Varianten wurden in das Turnier aufgenommen, da bei der Nachfrage, welche Variante eigentlich gemeint war, die Antwort „bitte beide verwenden“ zurück kam. Im Gegensatz zu „Dirk Alpha“ werden hier nur Runden gezählt, vor welchen der Gegner kooperiert hat. Bei jeder zehnten dieser Runden wird dann von „Dirk Beta“ eine Defektion eingestreut.

5.1.5.37. „Gartmann 1“**Autor:** Rüdiger Gartmann**Platz:** 52**Harte Attribute:** bleibend teilstatisch, deterministisch, bleibend teilperiodisch, Master (mit nicht bekanntem Slave; siehe unten, Erkennungssequenz ab Zug eins bis Zug sieben), mit Substrategie *ALLD* und *TFT*, zurückschlagend (nach PG474), im ersten Zug defektierend**Weiche Attribute:** kooperativ, verständlich**Nicht analysierbare Attribute:** keine

Diese Master-Strategie spielt in den ersten drei Zügen Defektion und danach eine bestimmte, zum Teil vom Gegner abhängige Sequenz. Nach den ersten sieben Zügen wird abhängig vom Gegner entweder *ALLD* oder *TFT* gespielt. Interessant ist, dass der eigene Slave anscheinend nicht erkannt wird, da dieser anfangs mit *CCCCD* beginnt und in den Runden zwei und drei keine Kooperation beim Slave vorkommen darf.

5.1.5.38. „Gartmann 2“**Autor:** Rüdiger Gartmann**Platz:** 51**Harte Attribute:** bleibend teilstatisch, deterministisch, bleibend teilperiodisch, Slave (mit nicht bekanntem Master; siehe unten, Erkennungssequenz ab Zug eins: *CCCCD*), mit Substrategie *ALLC* und *ALLD*, im ersten Zug kooperierend**Weiche Attribute:** aggressiv, wagemutig, total selbstmordgefährdet, verständlich, unkooperativer Slave**Nicht analysierbare Attribute:** keine

Diese Slave-Strategie spielt *ALLD* bei erkannter Anfangssequenz *?DDDCC* und spielt selbst am Anfang *CCCCD*. Danach wird entweder *ALLC* oder *ALLD* gespielt.

5.1.5.39. „Gralla50“

Autor: Rafael Gralla

Platz: 54

Harte Attribute: statisch, deterministisch, periodisch, im ersten Zug kooperierend

Weiche Attribute: aggressiv, total selbstmordgefährdet, verständlich, angepasst an die konkrete Spiellängenverteilung

Nicht analysierbare Attribute: keine

Diese eher einfache Strategie spielt im Wechsel immer 50 Züge Kooperation und 50 Züge Defektion. Dabei wird mit Kooperation begonnen. Das Konzept hinter der Strategie ist laut Autor die Hoffnung, dass der Gegner durch die anfänglichen 50 Züge Kooperation getäuscht wird, d. h. die Strategie als kooperierend einstuft und dann entweder in einen statischen Zustand übergeht oder, bei Majority-ähnlichem Verhalten des Gegners, die Strategie eher wohlwollend gegenüber tritt und die 50 Züge der Defektion ohne Gegenwehr über sich ergehen lässt. Die Periodenlänge wurde so gewählt, dass sie die Mindestrundenzahlen teilt (es ist jedoch nicht klar, warum).

5.1.5.40. „Dreiundzwanzig“

Autor: Matthias Miemczyk

Platz: 56

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.41. „Resistance is futile“

Autor: Christian Bredemeier

Platz: 57

Harte Attribute: dynamisch, randomisiert, periodisch, zurückschlagend (nach PG474), im ersten Zug kooperierend

Weiche Attribute: stark randomisiert, aggressiv, total selbstmordgefährdet, angepasst an die konkrete Spiellängenverteilung

Nicht analysierbare Attribute: keine

Diese Strategie hat eine Periodenlänge von 25 Zügen. In den ersten fünf Zügen wird immer kooperiert. Danach wird bis zum Ende der Periode grundsätzlich in jedem zweiten Zug, beginnend mit dem ersten folgenden, defektiert. Bei der verbleibenden Hälfte der Züge wird geprüft, ob der Gegner während der letzten fünf Züge genau zwei- oder dreimal defektiert hat. Ist dies der Fall, wird nur mit einer Wahrscheinlichkeit von $\frac{1}{3}$ defektiert, ansonsten wird grundsätzlich defektiert. Laut Autor ist der Sinn dahinter, dass durch die Defektion in der Hälfte der Züge im zweiten Periodenabschnitt die gegnerische Strategie gereizt werden soll. Sehr kooperative Strategien sollen dadurch, dass sie weniger als zweimal in den letzten fünf Zügen defektiert haben, erkannt und ausgenommen werden können.

Aggressivere Strategien hingegen sollen daran erkannt werden, dass sie in den letzten fünf Zügen mindestens viermal defektiert haben. In diesem Falle sei eine Kooperation aussichtslos, weswegen die Strategie auch hier defektiert. Ansonsten wird eine Kooperation mit einer Wahrscheinlichkeit von $\frac{2}{3}$ versucht. Die erste Periodenhälfte ist dazu gut, den Gegner am Anfang „milde“ zu stimmen und in den späteren Perioden die Erkennung nicht durch gegnerische Rache aus der letzten Periode zu gefährden. Die Periodenlänge wurde so gewählt, dass sie die Mindestundenlängen teilt.

5.1.5.42. „Delling“

Platz: 61

Autor: Jan Schultze

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.1.5.43. „Sklave“

Platz: 62

Autor: Christian Gentges

Harte Attribute: bleibend teilstatisch, deterministisch, bleibend teilperiodisch, Slave (mit Master: Meister, Erkennungssequenz ab Zug eins: DDCDCDDCCDCDC), mit Substrategie *ALLC* und *ALLD*, zurückschlagend (nach PG474), im ersten Zug defektierend

Weiche Attribute: aggressiv, wagemutig, verständlich, zurückschlagend (nach Axelrod), unkooperativer Slave

Nicht analysierbare Attribute: keine

Diese Strategie spielt in den 13 ersten Zügen eine Erkennungssequenz, die vom Master „Meister“ erkannt wird. Falls erkannt wird, dass der Gegner nicht der Master ist, wird immer defektiert, ansonsten ab Zug 14 immer kooperiert.

5.1.5.44. „Rasumichin“

Autor: Heiko Röglin

Platz: 63

Harte Attribute: nicht analysiert

Weiche Attribute: nicht analysiert

Nicht analysierbare Attribute: nicht analysiert

5.2. Ergebnistabelle des Turniers

Tabelle 5.1.: Ergebnistabelle zum Turnier auf dem Campusfest der Universität Dortmund (2005)

| Rang | Strategie Name | Punkte Gesamt |
|-------------|---|----------------------|
| 1 | <i>Meister</i> | 2993157 |
| 2 | <i>Netzer</i> | 2971664 |
| 3 | <i>MN1</i> | 2962952 |
| 4 | <i>Titfortwotats</i> | 2929551 |
| 5 | <i>Nils-TFT</i> | 2929493 |
| 6 | <i>S-TFT</i> | 2929423 |
| 7 | <i>SebrassiIPDII</i> | 2928422 |
| 8 | <i>Nils-TFT(2)</i> | 2925875 |
| 9 | <i>SpäteRache</i> | 2913842 |
| 10 | <i>THModGradual1</i> | 2909061 |
| 11 | <i>THModGradual2</i> | 2906714 |
| 12 | <i>TitForTatPlus1</i> | 2902501 |
| 13 | <i>Simon2</i> | 2885196 |
| 14 | <i>He-Man</i> | 2880267 |
| 15 | <i>Zweiundvierzig</i> | 2877407 |
| 16 | <i>SchwalliFaku</i> | 2873747 |
| 17 | <i>FJLastTwenTit for tatwithTfT-Recognition</i> | 2867911 |
| 18 | <i>Simon</i> | 2867329 |
| 19 | <i>FJLast TwenTit for tat</i> | 2857566 |
| 20 | <i>S-TFFT</i> | 2856911 |
| 21 | <i>SchwallisGradual</i> | 2847031 |
| 22 | <i>SebrassiIPDI</i> | 2845711 |
| 23 | <i>VerzögerteBestrafung</i> | 2839526 |
| 24 | <i>LateTFT</i> | 2832038 |
| 25 | <i>BorisKonrad1</i> | 2822942 |
| 26 | <i>BorisKonrad2</i> | 2821958 |
| 27 | <i>adamone</i> | 2820195 |
| 28 | <i>TBattleVersion4(2)</i> | 2819835 |
| 29 | <i>TBattleVersion4</i> | 2819740 |
| 30 | <i>w-Random</i> | 2815276 |
| 31 | <i>w-Random(2)</i> | 2802163 |
| 32 | <i>MugglingThrough(2)</i> | 2800898 |
| 33 | <i>MugglingThrough</i> | 2800628 |
| 34 | <i>eva Wobe Ga(2)</i> | 2785944 |
| 35 | <i>eva Wobe Ga</i> | 2776284,5 |
| 36 | <i>StrategyByBas(2)</i> | 2763096 |
| 37 | <i>StrategyByBas</i> | 2750760 |
| 38 | <i>DerZornGottes(2)</i> | 2743639 |
| 39 | <i>DerZornGottes</i> | 2738113 |
| 40 | <i>Sauer2</i> | 2733952 |
| 41 | <i>Sauer1</i> | 2729080 |
| 42 | <i>YoungCapone</i> | 2727824 |
| 43 | <i>GreenShark</i> | 2682136 |
| 44 | <i>Raskolnikoff</i> | 2646428 |
| 45 | <i>MajorEisberg</i> | 2594927 |
| 46 | <i>Throat</i> | 2510968 |

wird fortgesetzt

Fortsetzung von der letzten Seite

| Rang | Strategie Name | Punkte Gesamt |
|-------------|-----------------------------|----------------------|
| 47 | <i>adamtwo</i> | 2504708 |
| 48 | <i>MN2</i> | 2316143 |
| 49 | <i>DirkAlpha</i> | 2299867 |
| 50 | <i>DirkBeta</i> | 2042087 |
| 51 | <i>Gartmann2</i> | 1913418 |
| 52 | <i>Gartmann1</i> | 1832666 |
| 53 | <i>Gralla50(2)</i> | 1797750 |
| 54 | <i>Gralla50</i> | 1795064 |
| 55 | <i>Random</i> | 1748006,5 |
| 56 | <i>Dreiundzwanzig</i> | 1725472 |
| 57 | <i>Resistanceisfutile</i> | 1702214,5 |
| 58 | <i>DieMachtvonGrayskull</i> | 1689573 |
| 59 | <i>TastyFish</i> | 1684085 |
| 60 | <i>Betonschuh</i> | 1680868 |
| 61 | <i>Delling</i> | 1664193 |
| 62 | <i>Sklave</i> | 1645595 |
| 63 | <i>Rasumichin</i> | 1643014 |
| 64 | <i>Deep</i> | 1640406 |
| 65 | <i>MinorEisberg</i> | 1603729 |

Teil III.
Konferenzarbeiten

6. Einleitung

Als Novum sowohl für die Betreuer als auch für die Mitglieder dieser PG wurde am Anfang des zweiten Semesters beschlossen, zu zwei Konferenzen Paper einzusenden. Diese waren die GECCO 2006 und die CEC 2006. Die Paper sollten vornehmlich auf den Arbeiten des zweiten Semesters basieren und einen Hauptteil des Endberichtes darstellen. Insgesamt wurden vier Paper geschrieben, von denen zum Zeitpunkt der Fertigstellung dieses Endberichtes drei bereits eingesandt wurden, davon eins an die GECCO und zwei an die CEC. Das vierte Paper konnte aus Zeitmangel noch nicht fertig gestellt werden, die Untergruppe beschloss allerdings dieses Paper bei einer anderen Konferenz in den nächsten Monaten einzusenden.

In diesem Teil des Endberichtes werden die jeweiligen Untergruppen ihre Arbeit des zweiten Semesters vorstellen. In jedem Kapitel wird zunächst für jede Untergruppe kurz beschrieben, mit welchem Thema und aus welchen Gründen sich diese Untergruppe mit dem jeweiligen Thema auseinandersetzt. Anschließend wird das eingesandte Paper auf Deutsch zusammengefasst und in den folgenden Kapiteln werden dann die Ergebnisse ausführlicher dargelegt als es im Paper möglich war. Auch wird auf negative Ergebnisse eingegangen, welche nur in zweiter Linie Einfluss auf die produzierten Resultate der Gruppen hatten, doch für andere Personen, die sich mit der Materie beschäftigen, durchaus wissenswert sind.

Die erste Gruppe beschäftigte sich mit verschiedenen Umgebungen, welche in einem IPD-Turnier verwendet werden könnten. Ihr Hauptaugenmerk lag dabei auf dem Master-Slave-Problem und wie dieses umgangen werden kann und wie dadurch der Master-Slave-Vorteil minimiert werden kann. Die zweite Gruppe nutzte agentenbasierte Simulationen, um verschiedene Faktoren zu untersuchen, welche einen Einfluss auf die Entwicklung von Kooperation haben (wie zum Beispiel Bestrafung und Reputation). Die dritte Gruppe schließlich hat mit Hilfe evolutionärer Algorithmen neue Strategien entwickelt, welche in einem IPD-Turnier möglichst gut abschneiden sollten.

Am Ende dieses Abschnittes wird von allen Untergruppen ein Fazit gezogen, wobei vor allem auffällige Ergebnisse noch einmal zusammengefasst werden.

7. Alternative Turnierumgebungen

7.1. Einleitung

In einer Kleingruppe wurde früh erkannt, dass es keine allumfassende Umgebung für alle Problemstellungen geben kann. Hieraus erwuchs die Motivation, dass je nach Problemstellung eine eigene Umgebung definiert und implementiert werden sollte, potentiell sogar in jeder einzelnen Arbeitsgruppe.

Um festzustellen, wie gut Strategien im IPD abschneiden, wurde sich in den meisten Fällen auf einen paarweisen Vergleich der Strategien beschränkt, inklusive der eigenen Strategie. In diesen so genannten *Round-Robin*-Turnieren wurden dann am Ende einfach die Punkte aus allen Spielen addiert, wodurch sich eine Reihenfolge unter den Strategien ergab. Gute Strategien für eine Variante dieses *Round-Robin*-Turniers zeigen jedoch oft auch Schwächen gegen schon nur kleine Modifikationen dieser Umgebungsart. Dies hängt auch insbesondere von den anderen beteiligten Strategien ab.

Robert Axelrod hat 1981 zwei Computerturniere durchgeführt, um gute oder gar die beste Strategie zu finden. Als Modus wurde ein *Round-Robin*-Turnier veranstaltet, in dem jede Strategie einzeln jeweils 200 Züge gegen alle anderen, sich selbst und der zufälligen Strategie *Random* angetreten ist. Die Anzahl der Züge war den Autoren der Strategien vorher nicht bekannt. Der Sieger des Turniers war eine sehr verständliche Strategie namens „Tit for Tat“ (*TFT*). Diese Strategie kooperiert im ersten Zug und in jedem weiteren Zug spielt sie den Zug, den der Gegner beim letzten Mal benutzt hat. Das überraschende Ergebnis veranlasste Axelrod, das Turnier in größerem Rahmen zu wiederholen. Wiederum gewann *TFT*. Diese Strategie kann zwar nie im direkten Vergleich gewinnen (d. h. sie spielt entweder unentschieden oder sie verliert im paarweisen Spiel), aber sie hat durchschnittlich so die meisten Punkte erreicht.¹

Die Turniere im Rahmen der Konferenzen *CEC'2004* und der *CIG'05* - im Grunde Wiederholungen des Originalexperimentes von Axelrod (mit verschiedenen weiteren Optionen wie beispielsweise Rauschen) - führten zu Problemen mit der klassischen *Round-Robin*-Umgebung. Viele Teilnehmer nutzten das Turnier, um mehrere Strategien einzusenden, mit dem Ziel, mit wenigstens einer Strategie einen der vorderen Plätze zu belegen. Oft geschah dies durch das Ausnutzen von ebenfalls mitgeschickten Strategien, die sich selber aufopferten und vom selben Autor eingesendet wurden. Als Beispiel gewann das erste Turnier eine Strategie einer Forschungsgruppe der Universität Southampton, welche über 100 (von insgesamt 223) weitere Strategien als Unterstützung hatte.

Das von der PG474 im Jahre 2005 veranstaltete Turnier zeigte, dass es auf organisatorischem Weg kaum möglich ist, ein Zusammenwirken von Strategien in einer Master-Slave-Beziehung zu verhindern. Der Regelzusatz, nur zwei Strategien pro Teilnehmer zuzulassen, wurde von einigen Teilnehmern versucht zu hintergehen. Das Herausfiltern von „Betrügnern“ hat sich schon bei der geringeren Anzahl von Einsendungen als äußerst schwierig dargestellt. Somit stellt sich die Frage nach anderen Turnierumgebungen, die Master-Slave-Strategien erkennen und bestrafen können, falls sie nicht erwünscht sind.

In der Anfangsphase der Gruppenarbeit wurden deswegen existente Umgebungen gesucht und genauer erforscht. Erstaunlicherweise wurde bald festgestellt, dass vollkommen neue Umgebungen äußerst rar sind. Dafür existieren Untersuchungen von vielen Autoren über verschiedene *Payoff Varianten* des IPDs und damit verbundene Robustheitsanalysen der Ergebnisse von Axelrod (vgl. [Nac92],

¹Ausführliche Ergebnisse finden sich in „*Die Evolution der Kooperation*“ [Axe05], auf den Seiten 26-48 und 173-178

[Fog93], [Mue88]). Auch verschiedene Ansätze mit Fehlern in der Informationsübertragung (dem so genannten Rauschen) wurden relativ häufig betrachtet (vgl. [Axe05], [HC88] oder [Hof00]).

Änderungen der gesamten Turnierumgebung sind jedoch sehr selten untersucht worden. Zwar wurde schon in [HC88] eine Umgebung, in der Strategien aus dem Turnier ausscheiden können, präsentiert. Wird diese Art des *klassischen K.O. Systems* aber mit der *Round-Robin*-Variante verglichen, zeigt sich, dass sich hier keine Kooperation etablieren kann. Eine immer defektierende Strategie (*ALLD*) ist hier dominant und kann keinen Paarvergleich verlieren, wodurch sie immer siegreich ist. *Ausstiegsmöglichkeiten* wie in [DM95] sind zwar auch eine Umgebungsänderung, machen aber eine Anpassung der Strategien an diese Möglichkeit erforderlich. Würde diese Umgebung auf normale IPD-Strategiemengen angewendet werden, ergäbe sich keine Veränderung.

Deshalb war das Ziel, eine neue Turnierumgebung zu erschaffen, die die klassische *Round-Robin*-Variante komplett ersetzen kann und das IPD an sich nicht verändert bzw. erweitert. Der Schwerpunkt hierbei lag zudem in der Entwicklung und der Analyse von Umgebungen, die den Vorteil von Master-Slave-Strategien vermeiden oder verringern. Wichtig hierbei bleibt insbesondere, dass die Platzierungen der restlichen Strategien nicht wesentlich verschoben werden sollten.

Somit entstanden im Laufe der Zeit insgesamt vier Umgebungen für das IPD. Im ersten Semester der PG474 entstand eine Version eines klassischen *Round-Robin*-Turniers. Hierbei war im Regelwerk schon der Zusatz, nur zwei Strategien pro Teilnehmer zuzulassen, enthalten. In der weiteren Kleingruppenarbeit im zweiten Semester entstanden dann die neuen Umgebungen *Jerusalem* und *Jerusalem over all*. Als vierte Umgebung wurde die *ökologische Umgebung* von Axelrod [Axe05] implementiert, an die *Round-Robin*-Variante des PG-Turniers angepasst und untersucht.

7.2. Zusammenfassung „Environments for the IPD“

7.2.1. Die Problemstellung

Bei der Veranstaltung von IPD-Turnieren tritt die Frage auf, ob Kooperation zwischen Master und Slave erlaubt werden soll oder nicht. Wird diese Kooperation zugelassen, ist eine Beschränkung auf eine bestimmte Anzahl von Strategien pro Teilnehmer anzuraten, um das Turnier nicht alleine durch die Masse der Slaves dominieren zu lassen. Manche könnten auf den Gedanken kommen, dass Master-Slave-Strategien immer verboten werden sollten, doch darf nicht außer Acht gelassen werden, dass das IPD dadurch eine interessante Wendung in den letzten Jahren erhalten hat. IPD-Turniere spalten sich dadurch nun in zwei Bereiche, der eine mit Zusammenarbeit zwischen einzelnen Strategien, der andere nur mit Einzelkämpfern. Die Möglichkeiten für Strategien erweitern sich und der Schwerpunkt verlagert sich je nach Turnierumgebung. Im Master-Slave-Fall kommt es nun darauf an, statt einzelne Strategien ganze Strategiemengen zu entwickeln, die in ihrer Gesamtheit gut abschneiden.

Sollen Master-Slave-Strategien nicht erlaubt werden, tritt sofort das schwerwiegende Problem auf, dass solche wegen fehlender Kontrollmöglichkeiten in einem öffentlichen Turnier nicht organisatorisch verboten werden können.

7.2.1.1. Master-Slave-Strategien

Die sogenannten Master bzw. Slaves sind Strategien, die sich gegenseitig erkennen können. Dies geschieht dabei immer mit dem Ziel, den Master zu unterstützen, üblicherweise indem der Slave ständig kooperiert, während der Master defektiert, damit wird die optimale Gewinnausschüttung für den Master gewährleistet. Die Erkennung kann entweder einseitig oder auch beidseitig sein, im letzteren Fall würden Master- und Slave-Strategien sich gegenseitig erkennen. Als Erkennung kann natürlich nur die Folge von Kooperationen und Defektionen des aktuellen Spiels dienen, auf andere Informationen hat eine Turnierstrategie i. A. keinen Zugriff. Das Erkennen eines Slaves bedeutet nicht zwangsweise, dass der Master Defektionen in der Erkennungssequenz spielen muss. Haben

sich Master und Slave erkennt, wechselt der Master zur ständigen Defektion. Ausnahme ist, wenn der vermeintliche Slave einen nicht vorgesehenen Zug spielt, sich also zu einem späteren Zeitpunkt des Spielverlaufs gar nicht als die passende Slave-Strategie herausstellt. Analog dazu wechselt eine Slave-Strategie in einen Modus der ständigen Kooperation, zumindest bis die vermeintliche Master-Strategie einen Zug macht, der nicht vorausgesehen wurde.

Es wird nicht ausgeschlossen, dass der Master zur Erkennung z. B. einfach *TFT* spielt und dann in einer bestimmten Runde anfängt zu defektieren, falls der Gegenspieler sich nicht durch unpassende Züge verrät. Master-Strategien können konkret daran erkannt werden, dass sie in drei Zustände aufgeteilt werden können:

- Das Spielen einer vorgefertigten Sequenz zur Erkennung (evtl. auch 0 Züge lang)
- Ständige Defektion oder
- Eine Ausweichstrategie.

Slave-Strategien können analog dazu ebenfalls in drei Zustände aufgeteilt werden:

- Das Spielen einer vorgefertigten Sequenz zur Erkennung (evtl. auch 0 Züge lang)
- Ständige Kooperation oder
- Eine Ausweichstrategie.

Im Verlauf der Arbeit stellte sich die Frage, ob Slaves immer gegen alle anderen Strategien defektieren müssen. Manchmal ist es tatsächlich sinnvoller, wenn sie das nicht tun. In einem solchen Fall wird von einem *kooperativen Slave* gesprochen. *Kooperative Slaves* überleben länger als die nur defektierenden *unkooperativen Slaves*, allerdings unterstützen sie ihren Master auch nicht so massiv, weil sie nicht alle anderen Strategien durch Defektionen schädigen. Das kann sich in manchen Strategiemengen jedoch trotzdem für den Master auszahlen. *Kooperative Slaves* sind dabei schwerer zu erkennen, da sie in einer Grauzone zwischen normalen Strategien und Slaves einzuordnen sind. Deswegen wird erstmal von *unkooperativen Slaves* ausgegangen.

7.2.1.2. Unmöglichkeit, das Problem exakt zu lösen

Üblicherweise erfolgt die Punkteberechnung im Verlauf einzelner Turnierrunden additiv, wobei sämtliche erreichten Punkte positiv sind. So wird z. B. bei den Turnieren von Axelrod [Axe05] die Punktzahl einfach nach jedem Spiel aufsummiert. Wir gehen von folgender noch etwas schwächeren Aussage aus:

Eine Turnierumgebung besteht aus einzelnen Turnierrunden. Ferner sei jeder Slave ein *unkooperativer Slave*.

Eine Turnierumgebung heißt genau dann *monoton*, wenn gilt:

1. Alle Punkte aus Spielen, die in die Gesamtwertung eingehen, sind positiv und werden nach der Standard-Auszahlungsmatrix berechnet.
2. Die so berechneten Punkte werden mit einer streng monotonen Funktion f gewichtet und gehen dann so in die Gesamtpunktzahl ein, dass diese auch streng monoton wächst.

Normalerweise spiegelt die Funktion f , wie im obigen Beispiel, einfach die Identität wieder. Das entspricht dann einer einfachen Addition der Punkte wie z. B. in der Standardumgebung. Es ist aber durchaus möglich, dass die Punkte erst mit einer streng monoton steigenden Funktion f gewichtet und dann z. B. mit der Gesamtpunktzahl multipliziert werden. Auch kann die Funktion f veränderlich sein, z. B. abhängig von der aktuellen Turnierrunde, so dass immer weniger Punkte pro Runde vergeben werden. Dabei muss jedoch beachtet werden, dass f eine Strategie nicht generell schlechter

als eine andere bewerten darf, d. h. die Funktion f darf bei der Bewertung einer Strategie nur nach deren Punkten urteilen, nicht nach deren Name.

Bei *monotonen* Turnierumgebungen entwickeln sich die Punkte über die Turnierrunden damit streng monoton wachsend. Insbesondere enthält diese Klasse alle Turnierumgebungen, die wie die Axelrod-Umgebung additiv sind bei positiver Punktevergabe. Auch alle vorgeschlagenen neuen Umgebungen sind *monotone* Turnierumgebungen.

Es wird nun ein Master mit der äquivalenten Strategie ohne irgendeinen Slave verglichen, d. h. mit der gleichen Strategie, welche aber kein Master ist. Es ist jetzt nicht möglich, diesen Master immer so abschneiden zu lassen wie dieselbe Master-Strategie ohne Slaves. Der Master wird hier zumindest anfangs einen Vorsprung durch seine Slaves erhalten und diesen dann wegen der monoton wachsenden Punkteberechnung auch behalten. Bis jetzt wurde allerdings noch keine Annahme bezüglich der Strategiemenge gemacht, also wird als Strategie die Gewinnerstrategie gewählt und daraus das Negativresultat gefolgert: Für jede *monotone* Turnierumgebung gibt es eine Strategiemenge, in der ein Master gewinnt. Weiterhin gilt über alle Strategiemengen, dass alle Master-Strategien besser abschneiden als ihre äquivalenten Nicht-Master-Strategien.

Im Umkehrschluss bedeutet das, dass eine Turnierumgebung, die den Vorsprung von Master-Slave restlos tilgen möchte, nicht *monoton* nach obiger Definition sein darf. Das erscheint jedoch unnatürlich, da dann z. B. schon erreichte Punkte einer Strategie im weiteren Verlauf wieder zurückgenommen werden können. Bei *monotonen* Umgebungen, ist es also notwendig, unser Problem umzuformulieren:

Es wird eine Turnierumgebung gesucht, die den Vorsprung einer Master-Strategie durch seine Slaves so weit wie möglich minimiert, denn vermieden werden kann dieser Vorsprung nicht.

Bei kooperierenden Slaves wird das ganze Problem schwieriger, da die Slave-Strategien sich unter Umständen nur noch marginal von normalen Strategien unterscheiden. Es lässt sich aber vermuten, dass eine gute Turnierumgebung für unkooperative Slaves auch hierfür gut geeignet ist. Wie gut, hängt jedoch stark davon ab, wie sehr ein Slave kooperiert.

7.2.2. Neue Umgebungen

Insgesamt werden vier Umgebungen präsentiert: PG474, *Jerusalem*, *Jerusalem over all* und eine Variante der von Axelrod [Axe05] eingeführten *ökologischen Umgebung*.

7.2.2.1. PG 474 Umgebung

Zur Durchführung eines IPD-Turniers im Jahre 2005 entwickelte die PG474 der Universität Dortmund eine - im Gegensatz zur klassischen Axelrod Umgebung - modifizierte *Round-Robin*-Umgebung. Eine Zusammenarbeit zwischen einzelnen Strategien (im Sinne der Strategien von Southampton mit Kommunikation über Spielzüge) sollte zwar erlaubt sein, aber nicht durch eine Masse von Slave-Strategien spielentscheidend wirken. Deshalb durfte jeder Teilnehmer nur maximal zwei Strategien einreichen. Auf fehlende organisatorische Kontrollmöglichkeiten wird ausführlich in Abschnitt 7.3 eingegangen. Jede Strategie spielte hier gegen alle anderen Strategien und gegen sich selbst in drei verschiedenen Durchgängen. Die Anzahl der Runden in einem Durchgang wird zufällig ermittelt. Dabei gibt es eine Mindestrundenzahl von 100 Runden im ersten, 500 Runden im zweiten und 1000 Runden im dritten Durchgang. Vor jeder Folgerunde wird der Durchgang mit einer Wahrscheinlichkeit von $10/\text{Mindestrundenzahl}$ abgebrochen, also mit Wahrscheinlichkeiten von 0.1, 0.02 bzw. 0.01. Es wird also über die komplette Mindestrundenzahl gespielt und danach eine geometrisch verteilte Zufallsvariable addiert, wonach abgebrochen wird. Die konkreten Rundenzahlen werden vor dem Turnier ermittelt und sind für alle Spiele gleich. Die Gesamtpunktzahl einer Strategie für ein Turnier ist die Summe aller erzielten Punkte in allen Durchgängen. Um den Zufall gering zu halten wird das gesamte Turnier zehn mal durchgeführt und ein Mittelwert der Gesamtpunktzahlen jeder Strategie gebildet. Dieser Mittelwert ist Maß für die Reihenfolge der Strategien in der Rangliste. Sollte es

mehrere Platzierungen auf einem der Gewinnplätze geben, so entscheidet ein Stichturnier der beiden Strategien zusammen mit der Strategie *Random* über die Platzierung. Dieses Stichturnier wird so oft durchgeführt bis ein eindeutiges Ergebnis feststeht. Die andere Strategie rutscht entsprechend einen Platz in der Rangliste nach unten. Die Strategie mit den meisten Punkten erreicht den obersten Platz in der Rangliste.

7.2.2.2. *Ökologische Umgebung*

Bei dieser Turnierform wird zuerst für Populationen von Strategien die Fitness festgestellt. Dies geschieht durch die Durchführung eines einzelnen *Round-Robin*-Turniers. Die Fitness ist die Punktzahl, die eine Population von Strategien gegen die anderen Strategien der Population erreicht. Die Fitness einer Strategie bestimmt die Zusammensetzung der nächsten Generation (also im nächsten *Round-Robin*-Turnier). Am Anfang des Turniers ist der Anteil der Instanzen einer Strategie an der Gesamtpopulation gleich. Nach jeder Runde wird die neue Generation so zusammengesetzt, dass der Anteil einer Strategie proportional zu ihrer erreichten Punktzahl ist. Die Idee dahinter ist, dass in einer Population von interagierenden Individuen sich gute Strategien ausbreiten, d. h. die guten Strategien von anderen Individuen übernommen werden. Der Populationsanteil einer Strategie entspricht dann hier dem Populationsanteil der Individuen dieser Population, welche diese Strategie spielen. Da ein Individuum jeweils nur eine Strategie spielen kann, kann der Populationsanteil in der nächsten Generation wegen Rundung nicht genau proportional zur Summe der erzielten Punkte sein. In diesem Modell wird deshalb der Grenzfall der unendlich großen Population betrachtet und nur mit fraktionalen Werten gerechnet.

7.2.2.3. *Jerusalem over all*

Die Idee dieser Umgebung entstand in Anlehnung an das Spiel „Die Reise nach Jerusalem“, welches die Methode hinter dieser Umgebung auch treffend beschreibt:

Als Basis werden wieder einzelne *Round-Robin*-Turniere durchgeführt. Es wird jeweils die Strategie mit der geringsten Anzahl an bisher erreichten Punkten aus dem Turnier hinausgeworfen. Sie tritt dann in den nächsten Runden nicht mehr mit an. Dies wird nun so lange wiederholt bis nur noch eine Strategie übrig bleibt oder alle Strategien eine identische Punktzahl haben, so dass kein eindeutiger letzter Platz ermittelt werden kann. Letzteres tritt insbesondere dann in Kraft, wenn alle übrigen Strategien miteinander kooperieren. Jede Strategie erhält alle Punkte, die sie bis zu dem aktuellen *Round-Robin*-Durchgang erworben hat. Bei dieser Art von Umgebung ist es den Strategien möglich, sich ein gewisses Punktepolster aufzubauen, von dem sie in den späteren Runden zehren können.

7.2.2.4. *Jerusalem*

Bei dieser Variante der *Jerusalem*-Umgebung wird die Strategie mit den wenigsten Punkten des aktuellen *Round-Robin*-Durchgangs hinausgeworfen, es werden für die Ausscheidung nur die Punkte der aktuellen Runde betrachtet. Dies ist der entscheidende Unterschied zu der *Jerusalem over all*-Umgebung, denn hier kann ein potentiell vorher erwirtschaftetes Punktepolster nicht benutzt werden. Die Gesamtpunktzahl einer Strategie wird erst additiv berechnet nachdem das komplette Turnier vorbei ist.

7.2.3. Ergebnisse

Tabelle 7.1 zeigt die Ergebnisse aus allen Turnieren. Hierbei werden die Ergebnisse aus dem *Round-Robin*-Turnier (Spalte *B*) auf Basis der Umgebung PG474 mit den anderen Turnieren verglichen. Die Abweichung der Ränge in den anderen Umgebungen wird in den Spalten *C–E* dargestellt. \pm bedeutet, dass die Strategie *c* Ränge besser oder schlechter ist als im PG474 Turnier. Spalte *C*

Tabelle 7.1.: Turnierergebnisse aus allen Turnieren: Verglichen mit den Ergebnissen des PG474 Turniers (Spalte *B*), wird die Abweichung der Ränge in den anderen Umgebungen in den Spalten *C–E* dargestellt. \pm bedeutet, dass die Strategie *c* Ränge besser oder schlechter ist als im PG474 Turnier. Spalte *C* beschreibt die *ökologische Umgebung*, Spalte *D* *Jerusalem* und *E* *Jerusalem over all*. In Spalte *A* werden die einzelnen Strategien klassifiziert.

| A | B | C | D | E |
|--|-----|-----|-----|-----|
| (Master) <i>Netzer</i> | 1 | 0 | 0 | 0 |
| (Master) <i>Meister</i> | 2 | -3 | -10 | -9 |
| (Gradual variant) <i>ThmodGradual2</i> | 3 | -4 | -4 | -10 |
| (Gradual variant) <i>ThmodGradual1</i> | 4 | +2 | +2 | +2 |
| (Gradual variant) <i>Simon2</i> | 5 | -6 | +2 | +2 |
| (Master) <i>MN1</i> | 6 | +3 | +2 | +2 |
| (TfT variant) <i>Sebrassi IPD II</i> | 7 | +3 | +2 | +2 |
| (Gradual variant) <i>Schwalli Faku</i> | 8 | -5 | 0 | +1 |
| (Gradual variant) <i>Simon</i> | 9 | 0 | +3 | +3 |
| (standard strategy) <i>Gradual</i> | 12 | -3 | +1 | +3 |
| (standard strategy) <i>Tit for Tat</i> | 14 | -5 | 0 | -1 |
| (Master) <i>Zweiundvierzig</i> | 20 | -21 | -13 | -23 |
| (Master) <i>He-Man</i> | 26 | -17 | -8 | -19 |
| (Master) <i>Young Capone</i> | 40 | -14 | -10 | -14 |
| (Master) <i>Raskolnikoff</i> | 48 | -6 | -4 | -6 |
| (Master) <i>Green Shark</i> | 50 | -3 | 0 | -1 |
| (Master) <i>Major Eisberg</i> | 52 | -6 | -2 | -4 |
| (Master) <i>Throat</i> | 53 | -6 | -4 | -5 |
| ... | ... | ... | ... | ... |
| (Slave) <i>MN1</i> | 59 | +2 | 0 | 0 |
| (standard strategy) <i>Joss</i> | 66 | -3 | -2 | -2 |
| (standard strategy) <i>Random</i> | 70 | +2 | +1 | +1 |
| (Slave) <i>Zweiundvierzig</i> | 71 | -2 | -1 | -1 |
| (Slave) <i>He-Man</i> | 72 | -3 | -1 | -2 |
| (Tester variant) <i>Resistance is futile</i> | 73 | +1 | +2 | +2 |
| (Slave) <i>Green Shark</i> | 74 | 0 | 0 | -1 |
| (Slave) <i>Young Capone</i> | 75 | -2 | 0 | -1 |
| (standard strategy) <i>Per nasty</i> | 76 | +5 | 0 | +3 |
| (Slave) <i>Raskolnikoff</i> | 77 | -1 | 0 | 0 |
| (Slave) <i>Netzer</i> | 78 | -4 | 0 | 0 |
| (Slave) <i>Throat</i> | 79 | -1 | 0 | 0 |
| (Slave) <i>Major Eisberg</i> | 80 | +4 | 0 | 0 |
| (Slave) <i>Meister</i> | 81 | +2 | 0 | 0 |
| (standard strategy) <i>ALLD</i> | 82 | +1 | 0 | 0 |

beschreibt die *ökologische Umgebung*, Spalte *D Jerusalem* und *E Jerusalem over all*. In Spalte *A* werden die einzelnen Strategien klassifiziert.

In der Tabelle werden nur ausgewählte Strategien dargestellt. Der Fokus liegt auf den Master-Slave-Strategien und ihrer Abweichung in den jeweiligen Umgebungen. Es ist ersichtlich, dass die ersten beiden Strategien Master-Strategien sind. Die Ränge drei bis fünf sind Varianten der Standardstrategie *GRADUAL*, die sich im Wesentlichen nur in der Berechnungsweise des Defektionsgrades vom Original unterscheiden.

Die beste originale Standardstrategie im PG474 Turnier ist *GRADUAL* auf Platz 12. *TFT* landet nur auf Platz 14. Die Slave-Strategien schneiden im Allgemeinen sehr schlecht ab. Alle neun unkooperativen Slaves sind auf den Plätzen 71–81 von insgesamt 82 Plätzen zu finden. Der einzige kooperative Slave ist auf Platz 59 platziert.

In den Spalten *D* und *E* sind die Resultate von *Jerusalem* und *Jerusalem over all* zu finden. In beiden bleibt die Master-Strategie *Netzer* auf dem ersten Rang. Der Master auf dem zweiten Platz verliert aber in beiden Varianten einige Plätze (neun bzw. zehn). Im Allgemeinen sinken acht von insgesamt zehn Master-Strategien in der Platzierung signifikant ab. Die Slaves behalten ihre Plätze und bleiben auf den unteren Plätzen (Ränge 59–82). Ähnliche Ergebnisse werden in der *ökologische Umgebung* erzielt (Spalte *C*), obwohl die Reihenfolge der Slaves auf den unteren Plätzen etwas untereinander variiert. Die Ausnahme in den *Jerusalem* Umgebungen und der *ökologischen Umgebung* ist die Strategie *MN1*. Dies ist die einzige Master-Strategie die in allen Umgebungen Plätze gut macht und nach oben steigt. Das ist erklärbar durch ihren Slave *Slave MN1*, der der einzige kooperative Slave ist und sich somit länger dem Ausscheiden widersetzen kann.

7.2.4. Analyse der Ergebnisse

Die Umgebungen PG474, *Jerusalem*, *Jerusalem over all* und die *ökologische Umgebung* gehören alle in die Klasse der *monotonen* Turnierumgebungen. Es kann also für diese Umgebungen gefolgert werden, dass eine Strategie immer schlechter ist, als die identische Strategie, die einen Master Bonus hat. Die drei besten Master-Strategien *Netzer*, *Meister* und *MN1* spielen als alternative Strategie entweder eine Form von *GRADUAL* oder *TFT*. Dementsprechend schneiden sie auch in allen Fällen besser ab, als diese entsprechenden Standardstrategien. Die Slaves belegen weiterhin die letzten Plätze. Unkooperative Slaves schneiden schlechter ab als kooperative, die viel länger im Turnier bleiben, weil sie selber durch Kooperation Punkte machen und damit nicht als Slave erkennbar sind. Gute Strategien können als freundlich und zurückschlagend, oder als Master-Strategien klassifiziert werden. Aggressive Strategien, die keine Master sind, belegen durchgehend die unteren Plätze.

Es fällt auf, dass eine einzige Strategie noch schlechter als alle Slaves abschneidet, nämlich *Always Defect (ALLD)*, die immer defektierende Strategie. Eigentlich müsste diese Strategie mehr Punkte erwirtschaften als alle unkooperativen Slaves, da diese zwar gegen den überwiegenden Teil der Strategien auch defektieren, aber wenn sie auf ihren Master treffen, noch schlechter spielen. Der Grund liegt in der *Payment Matrix*, die gegenseitige Defektion mit einem Punkt belohnt, jedoch ein Ausbeuten mit 0 Punkten bestraft. Dass *ALLD* dennoch schlechter spielt, liegt an der Art und Weise der Slaves: sie müssen eine Erkennungsstrategie spielen, diese besteht notwendigerweise auch aus Kooperationen, sonst könnte sie nicht von *ALLD* unterschieden werden. Diese Erkennungssequenzen sind in der Regel zwar nicht länger als 5-20 Züge, allerdings werden sie gegen sämtliche 82 Strategien gespielt, denn jede Strategie muss auf den Master getestet werden.

In den *Jerusalem*-Umgebungen werden die unkooperativen Slaves schon früh im Turnier entfernt, weswegen die zugehörigen Master bestraft werden. Das ist genau das, was bei Turnieren, wo Master-Slave-Strategien verboten sind, erreicht werden soll. Natürlicherweise ändert sich die Strategiemenge im Laufe des Turniers. Die Strategiemenge wird sich immer weiter zu einem kooperativen Verhalten bewegen. Das mündet schließlich darin, dass alle übrig gebliebenen Strategien (in Spezialfällen dann nur eine einzige) miteinander kooperieren. Daran können leicht die Ergebnisse verifiziert werden, dass

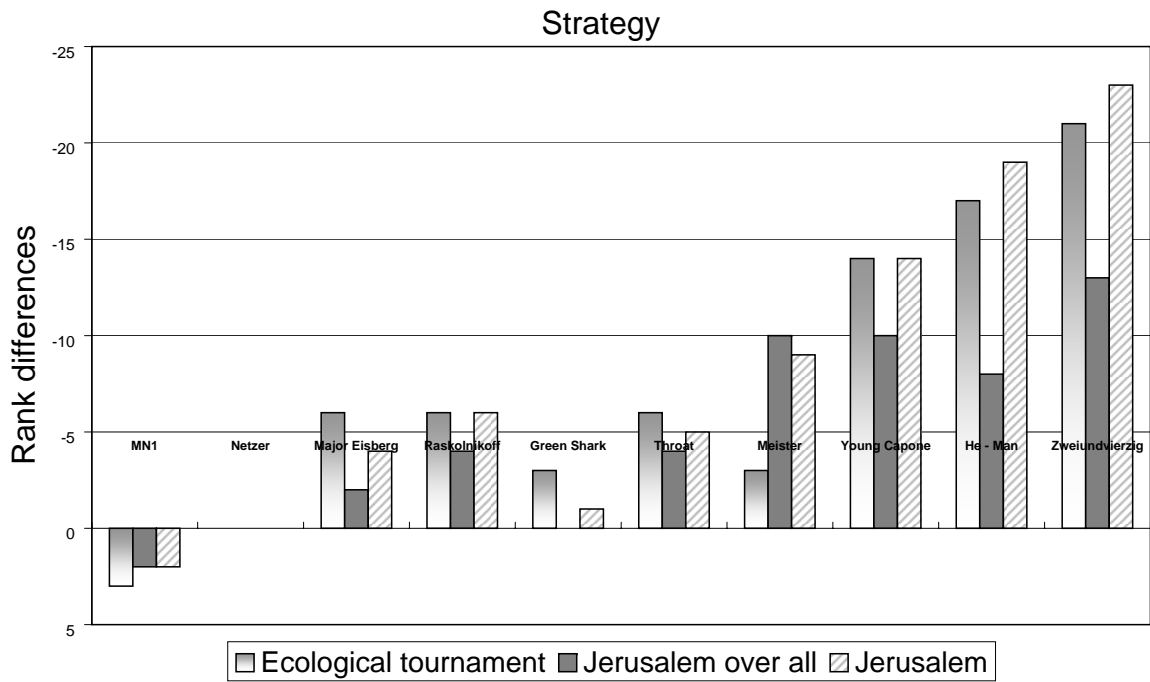


Abbildung 7.1.: Die Platzunterschiede aller zehn Master-Strategien in unseren verschiedenen Umgebungen im Vergleich mit der *Round-Robin*-Umgebung. Alle untersuchten Umgebungen mindern den Vorteil der Master-Strategien. Nur eine Strategie verbessert sich in diesen neuen Umgebungen: Die Strategie *Slave MN1*, denn sie ist ein sogenannter *kooperativer Slave*. Insgesamt bewirkt die *Jerusalem*-Umgebung den größten Effekt und ist deswegen für Turniere gut geeignet, in denen keine Master-Slave-Strategien erlaubt sind.

aggressive Strategien, solange sie keine Master-Strategien sind, schlecht abschneiden. Dadurch, dass immer die letzten Strategien entfernt werden, ist die Robustheit bei wiederholten Läufen derselben Turniersituation bei den unteren Rängen höher. Trotzdem ist die Robustheit aller Umgebungen so groß, dass auch auf den ersten Plätzen kaum eine Variation bemerkbar ist.

So hatten beispielsweise über 80% der Strategien der *Jerusalem*-Umgebung eine Standardabweichung bezüglich der Plätze von weniger als eins über den zehn Iterationen. Bei der *Jerusalem overall*-Umgebung waren es 75%. Die durchschnittliche Standardabweichung über die Plätze betrug 0,717 und 0,661 Plätze in den Umgebungen.

In der *ökologischen Umgebung* wurden 1200 Generationen berechnet, bis nur noch marginale Änderungen in der Populationsgröße aufgetreten sind. Abbildung 7.2 zeigt die Populationsgröße über die Generationen. Links unten im Bild gehen die Populationsgrößen aller unkooperativen Slaves und *ALLD* schon in den ersten Generationen gegen 0.

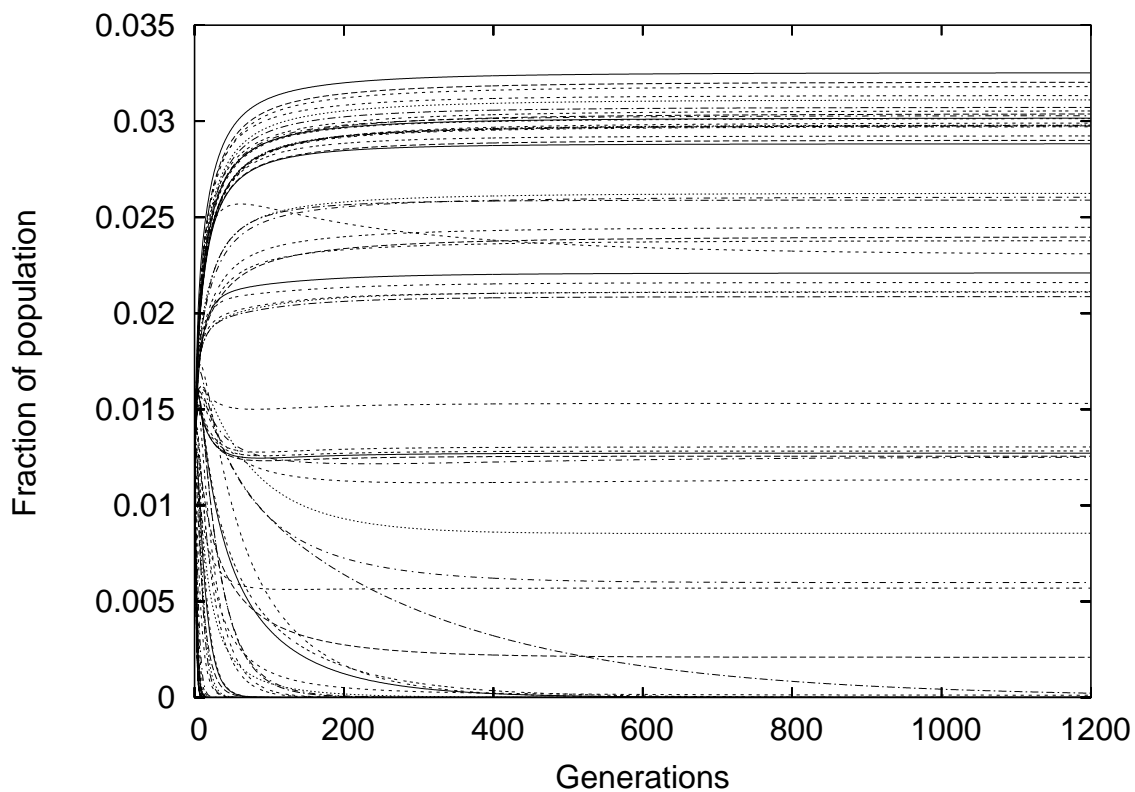


Abbildung 7.2.: Alle 82 Strategien in der *ökologischen Umgebung* über 1200 Generationen. Die Population der Slave-Strategien sinkt schon in den ersten Generationen gegen 0. Nach 1200 Generationen gibt es keine signifikanten Änderungen mehr in den Populationsanteilen, so dass abgebrochen wird.

Auf den untersten Plätzen treten ein paar kleine Differenzen im Vergleich zu beiden *Jerusalem*-Umgebungen auf. Das wird dadurch bedingt, dass hier die Strategien nicht diskret entfernt werden, und sich somit noch im späteren Verlauf beeinflussen können. Zum Beispiel gewinnt die Strategie *ALLD* einen Rang, weil sie die Chance hat, bei den zunehmend kooperativeren Populationsanteilen (z. B. *ALLC*) auch im späteren Verlauf des Turniers Punkte zu holen.

7.3. Weitere Ergebnisse

7.3.1. Gesamtergebnis

Im Paper konnten nicht alle Ergebnisse aus dem zweiten Semester verarbeitet werden. Die vollständige Vergleichstabelle aller erwähnten Umgebungen, also der Vergleich zwischen der *Round-Robin*-Umgebung PG474, den Umgebungen *Jerusalem* und *Jerusalem over all* und der *ökologischen Umgebung* findet sich in Tabelle 7.2. Strategien mit Anhang „(2)“ spiegeln dort Strategieduplikate wieder, die entstanden sind, damit einzeln eingesandte Strategien nicht Teilnehmern mit zwei eingesandten Strategien unterlegen sind. Dies geht auf die Restriktion zurück, dass beim PG474-Turnier nur zwei Strategien eingesandt werden durften. Im Paper wurde hauptsächlich diese Auswahl von Strategien als Grundlage für Resultate des *Round-Robin*-Turniers genommen, da dort der Effekt der Zusammenarbeit von Master-Slave-Strategien am geringsten ist. Notwendig ist diese Einschränkung jedoch nicht, es ist eine beliebige Anzahl von Slaves möglich. Aussagen im Paper, dass Master-Slave-Strategien viele Punkte erwirtschaften, könnten somit durch Zulassen von mehr Slaves beliebig ausgeweitet werden. Sobald schon nur ein Slave mit eingereicht wird, kann aufgrund des Negativresultates nicht mehr erwartet werden, dass unerlaubte Zusammenarbeit verhindert werden kann. Bei mehreren Slaves verändert sich das Negativresultat natürlich nicht, im Gegenteil ist dann noch nicht einmal zu erwarten, dass der Bonus des Masters mit den neuen Umgebungen signifikant gesenkt werden kann. Das ist jedoch eine unvermeidliche Aussage, so sind z. B. Strategiemengen mit einer *TFT*-Strategie und beliebig vielen Master-Slave-Strategien möglich. In solchen Strategiemengen erhöht sich natürlich die Chance für Slaves, den Master erheblich zu fördern.

Tabelle 7.2.: Ergebnistabelle von 82 Strategien in den Turnieren PG474, *ökologische Umgebung*, *Jerusalem over all* und *Jerusalem*. Strategien mit Anhang (2) spiegeln Strategieduplikate wieder, die entstanden sind, damit einzeln eingesandte Strategien nicht Teilnehmern mit zwei eingesandten Strategien unterlegen sind. Dies geht auf die Restriktion zurück, dass beim PG474-Turnier nur zwei Strategien eingesandt werden durften.

| | PG474 | Ökologische | Jerusalem over all | Jerusalem |
|----|------------------|--------------------|---------------------------|------------------|
| 1 | Netzer | Netzer | Netzer | Netzer |
| 2 | Meister | THModGradual1 | THModGradual1 | THModGradual1 |
| 3 | THModGradual2 | MN1 | Simon2 | Simon2 |
| 4 | THModGradual1 | Sebrassi IPD II | MN1 | MN1 |
| 5 | Simon2 | Meister | Sebrassi IPD II | Sebrassi IPD II |
| 6 | MN1 | TitForTat Plus1 | Simon | Simon |
| 7 | Sebrassi IPD II | THModGradual2 | THModGradual2 | Schwalli Faku |
| 8 | Schwalli Faku | Sebrassi IPD I | Schwalli Faku | Gradual |
| 9 | Simon | Simon | Sebrassi IPD I | SchwallisGradual |
| 10 | Sebrassi IPD I | SchwallisGradual | Gradual | Sebrassi IPD I |
| 11 | Tit for two tats | Simon2 | SchwallisGradual | Meister |
| 12 | Gradual | Späte Rache | Meister | TitForTat Plus1 |
| 13 | Nils-TFT | Schwalli Faku | TitForTat Plus1 | THModGradual2 |
| 14 | Tit-for-tat | Nils-TFT (2) | Tit-for-tat | S-TFT |
| 15 | SchwallisGradual | Gradual | Nils-TFT (2) | Tit-for-tat |
| 16 | FJ Last_Twen2 | Nils-TFT | Tit for two tats | Nils-TFT |
| 17 | S-TFT | S-TFT | Nils-TFT | Nils-TFT (2) |

wird fortgesetzt...

... Fortsetzung von der letzten Seite

| PG474 | Ökologische | Jerusalem over all | Jerusalem |
|--------------------------|-----------------------|-----------------------|-----------------------|
| 18 Nils-TFT (2) | Tit for two tats | S-TFT | Tit for two tats |
| 19 Späte Rache | Tit-for-tat | Späte Rache | Späte Rache |
| 20 Zweiundvierzig | FJ Last_Twen | FJ Last_Twen | FJ Last_Twen |
| 21 TitForTat Plus1 | FJ Last_Twen2 | FJ Last_Twen2 | FJ Last_Twen2 |
| 22 FJ Last_Twen | Boris Konrad 1 | Boris Konrad 1 | Boris Konrad 1 |
| 23 Boris Konrad 1 | Boris Konrad 2 | Boris Konrad 2 | Boris Konrad 2 |
| 24 Boris Konrad 2 | LateTFT | StrategyByBas | StrategyByBas (2) |
| 25 Mugging Through (2) | StrategyByBas (2) | StrategyByBas (2) | StrategyByBas |
| 26 He - Man | Mugging Through | Mugging Through | Mugging Through (2) |
| 27 Mugging Through | Mugging Through (2) | Mugging Through (2) | Mugging Through |
| 28 evaWobeGa | StrategyByBas | LateTFT | LateTFT |
| 29 evaWobeGa (2) | Hard Tit-for-Tat | Hard Tit-for-Tat | Slow Tit-for-Tat |
| 30 Hard Tit-for-Tat | Slow Tit-for-Tat | Slow Tit-for-Tat | S-TFTT |
| 31 Slow Tit-for-Tat | Tit-for-two-tats | S-TFTT | Tit-for-two-tats |
| 32 TBattle Version 4 | S-TFTT | Tit-for-two-tats | Hard Tit-for-Tat |
| 33 S-TFTT | Go by majority | Zweiundvierzig | Go by majority |
| 34 Tit-for-two-tats | Der Zorn Gottes (2) | He - Man | w-Random (2) |
| 35 TBattle Version 4 (2) | Sauer2 | Go by majority | w-Random |
| 36 LateTFT | Sauer1 | w-Random | Verzögerte Bestrafung |
| 37 adamone | w-Random | w-Random (2) | Der Zorn Gottes |
| 38 Young Capone | Der Zorn Gottes | adamone | Der Zorn Gottes (2) |
| 39 w-Random | w-Random (2) | evaWobeGa (2) | Sauer2 |
| 40 Go by majority | Verzögerte Bestrafung | Verzögerte Bestrafung | Sauer1 |
| 41 w-Random (2) | Zweiundvierzig | evaWobeGa | TBattle Version 4 |
| 42 StrategyByBas | adamone | TBattle Version 4 | TBattle Version 4 (2) |
| 43 Der Zorn Gottes | He - Man | TBattle Version 4 (2) | Zweiundvierzig |
| 44 Der Zorn Gottes (2) | Always cooperate | Der Zorn Gottes (2) | Always cooperate |
| 45 StrategyByBas (2) | TBattle Version 4 | Der Zorn Gottes | He - Man |
| 46 Verzögerte Bestrafung | Pavlov | Sauer2 | adamone |
| 47 Sauer2 | TBattle Version 4 (2) | Sauer1 | Pavlov |
| 48 Raskolnikoff | evaWobeGa (2) | Young Capone | evaWobeGa |
| 49 Sauer1 | Spite | Spite | evaWobeGa (2) |
| 50 Green Shark | evaWobeGa | Green Shark | Spite |
| 51 Spite | adamtwo | Pavlov | Green Shark |
| 52 Major Eisberg | Young Capone | Raskolnikoff | Young Capone |
| 53 Throat | Green Shark | adamtwo | adamtwo |
| 54 Mistrust | Raskolnikoff | Major Eisberg | Raskolnikoff |
| 55 Pavlov | Mistrust | Mistrust | Mistrust |
| 56 adamtwo | Hard majority | Always cooperate | Major Eisberg |
| 57 Hard majority | MN2 | Throat | Hard majority |

wird fortgesetzt...

... Fortsetzung von der letzten Seite

| | PG474 | Ökologische | Jerusalem over all | Jerusalem |
|----|----------------------|----------------------|---------------------------|----------------------|
| 58 | Always cooperate | Major Eisberg | Hard majority | Throat |
| 59 | MN2 | Throat | MN2 | MN2 |
| 60 | Dirk Alpha | Dirk Alpha | Dirk Alpha | Dirk Alpha |
| 61 | Prober | Prober | Prober | Prober |
| 62 | Dirk Beta | Gartmann2 | Gartmann2 | Gartmann2 |
| 63 | Gartmann2 | Gralla 50 | Dirk Beta | Dirk Beta |
| 64 | Tester | Gralla 50 (2) | Tester | Tester |
| 65 | Per kind | Per kind | Gralla 50 | Gralla 50 |
| 66 | Joss | Tester | Per kind | Gralla 50 (2) |
| 67 | Gralla 50 | Dirk Beta | Gralla 50 (2) | Per kind |
| 68 | Gralla 50 (2) | Random | Joss | Joss |
| 69 | Gartmann1 | Joss | Random | Random |
| 70 | Random | Gartmann1 | Gartmann1 | Gartmann1 |
| 71 | Dreiundzwanzig | Per nasty | Resistance is futile | Resistance is futile |
| 72 | Grayskull | Resistance is futile | Dreiundzwanzig | Dreiundzwanzig |
| 73 | Resistance is futile | Dreiundzwanzig | Grayskull | Per nasty |
| 74 | Tasty Fish | Tasty Fish | Tasty Fish | Grayskull |
| 75 | Betonschuh | Grayskull | Betonschuh | Tasty Fish |
| 76 | Per nasty | Minor Eisberg | Per nasty | Betonschuh |
| 77 | Rasumichin | Betonschuh | Rasumichin | Rasumichin |
| 78 | Delling | Rasumichin | Delling | Delling |
| 79 | Deep | Sklave | Deep | Deep |
| 80 | Minor Eisberg | Deep | Minor Eisberg | Minor Eisberg |
| 81 | Sklave | Always defect | Sklave | Sklave |
| 82 | Always defect | Delling | Always defect | Always defect |

Die angesprochene Betrachtung des PG474-Turniers bringt ein Problem mit sich: Es wurde im Paper gezeigt, dass eine organisatorische Vermeidung von Master-Slave-Strategien praktisch nicht möglich ist. Die Einschränkung auf zwei Strategien pro Teilnehmer fällt unter denselben Aspekt und es stellt sich die Frage, wie dieses sichergestellt werden konnte. Die Antwort lautet, dass in der Tat Probleme mit völlig identischen Slaves auftraten, die über fünf verschiedene Personen über unterschiedliche E-Mails eingeschickt wurden. Diese Slaves hätten ihren Master natürlich vielmehr gefördert, als lediglich der erlaubte eine Slave. Sie waren völlig identisch, so dass einfach ein Prüfen des Sourcecodes sicherstellen konnte, dass diese fünf zusätzlichen Teilnehmer genau den gleichen Slave eingesandt haben und aus dem Turnier ausgeschlossen wurden. Allerdings muss ein solcher Fall nicht so einfach erkannt werden können. Es gibt Methoden, diese Überprüfung zu umgehen, womit es dann in der Praxis wirklich nicht mehr möglich ist, Master-Slave-Strategien auszuschließen. Eine Überprüfung des Sourcecodes reicht nicht aus, da dieser trotz gleicher Funktion beliebig lang und komplex realisiert werden könnte. Zusätzlich kann mit Wahrscheinlichkeiten gearbeitet werden, so dass der Master nur mit einer großen Wahrscheinlichkeit unterstützt wird. Auf weitere Methoden soll im Folgenden noch einmal ausführlich eingegangen werden.

7.3.2. Entstehung der *Jerusalem*-Umgebungen

Die ersten Versuche mit der *Jerusalem* Umgebung fanden noch komplett ohne Punktzählung über Runden statt. Die Idee war es nach *Anzahl der Strategien* –1 Runden einen eindeutigen Sieger zu bekommen. Der letzte Platz des Turniers sollte einfach eliminiert werden, so dass am Ende genau eine Strategie übrig bleibt. Das erste Ergebnis, das mit unseren 82 Teststrategien auftrat, war etwas ernüchternd. Nachdem gut die Hälfte der Strategien ausgeschieden waren, kooperierten die übrigen Strategien nur noch und so konnte keine Strategie mehr ausscheiden. Alle Strategien erhielten dieselbe Punktzahl. Auf der anderen Seite war dieses Ergebnis auch als positiv zu bewerten, da erreicht werden konnte, dass unter den Gewinnern keine Slave-Strategie zu finden ist. Die Master-Strategien waren auch nicht die alleinigen Sieger, sondern fanden sich in einer großen Siegergruppe wieder. Diese erste Turnierform könnte z. B. als eine Art Vorentscheidung genutzt werden, wobei dann alle Slaves aus dem Turnier entfernt werden. Zu dem neuen Teilnehmerfeld könnten dann Standardstrategien beigefügt und ein neues *Round-Robin*-Turnier gestartet werden. So wären die Slave-Strategien effizient eliminiert.

Die weiteren Überlegungen führten zu den Gedanken wie die Punkte über die Runden gezählt werden können, um eine eindeutige Rangfolge unter den Strategien zu erhalten. Die erste Variante war, dass die Strategien jeweils die Punkte, die sie in einer Runde erzielt haben, direkt auf ihr Punktekonto gut geschrieben bekommen. So entstand die Variante *Jerusalem over all*.

Diese Variante sorgte nun für eine eindeutige Rangordnung unter den Strategien. Slave-Strategien scheiden in der ersten Runde aus. So erhalten die Master nur einen geringen Bonus aus den ersten Runden, der ihnen jedoch gegenüber anderen Strategien ohne Slaves einen Vorteil bietet. Der vermeintliche Nachteil dieser Variante ist, dass sich die Master-Strategien ein Polster in den ersten Runden aufbauen können und so in den späteren Runden nicht aus dem Turnier ausscheiden, auch wenn sie in einer Runde vielleicht sehr schlecht abschneiden.

Die Möglichkeit, sich ein Punktepolster aufzubauen, wurde in der Variante *Jerusalem* beseitigt. Der Gedanke hierbei war, das Ausscheiden nur über die aktuell letzte Runde zu berechnen. Die Punktezählung erfolgt am Ende des gesamten Turniers, wo jede Strategie die Punkte aus ihren jeweiligen Runden bekommt. So ergibt sich eine eindeutige Rangordnung unter den Strategien. Schlechte Master-Strategien scheiden jetzt aufgrund einer schlechten Runde aus dem Turnier aus und können nicht mehr von ihrem Slave profitieren. Jedoch erhalten Master-Strategien in der abschließenden Punktezählung einen Bonus aus den ersten Runden und sind somit wieder besser als identische Strategien ohne Slaves.

7.3.3. Ergebnisse der ökologischen Umgebung

Tabelle 7.3.: Ergebnistabelle von 82 Strategien in der ökologische Umgebung. Es wurden insgesamt zehn Läufe durchgeführt. Die Standardabweichung der Plätze ist bei über 50% der Strategien kleiner als eins und bei über 80% der Strategien kleiner als zwei. Im Durchschnitt betrug sie 0,717. Somit ist auch diese Umgebung robust gegenüber Mehrfachiterationen.

| Platz | Strategienname | Mittelwert über Plätze | Standardabweichung über Plätze |
|-------|-----------------|------------------------|--------------------------------|
| 1 | Netzer | 1,7 | 0,9 |
| 2 | THModGradual1 | 2,4 | 2,11 |
| 3 | MN1 | 3,4 | 0,66 |
| 4 | Sebrassi IPD II | 4,6 | 1,2 |
| 5 | Meister | 6 | 1,18 |
| 6 | TitForTat Plus1 | 7,5 | 1,36 |

wird fortgesetzt...

... Fortsetzung von der letzten Seite

| Platz | Strategie Name | Mittelwert über Plätze | Standardabweichung über Plätze |
|-------|-----------------------|------------------------|--------------------------------|
| 7 | THModGradual2 | 9 | 9,55 |
| 8 | Sebrassi IPD I | 9,8 | 1,66 |
| 9 | Simon | 10,3 | 5,27 |
| 10 | SchwallisGradual | 12 | 3,77 |
| 11 | Simon2 | 12,1 | 6,19 |
| 12 | Späte Rache | 13 | 2,53 |
| 13 | Schwalli Faku | 13,5 | 5,77 |
| 14 | Nils-TFT (2) | 13,6 | 1,43 |
| 15 | Gradual | 14,5 | 4,88 |
| 16 | Nils-TFT | 15,1 | 1,81 |
| 17 | S-TFT | 15,4 | 1,43 |
| 18 | Tit for two tats | 15,6 | 1,36 |
| 19 | Tit-for-tat | 16,1 | 1,92 |
| 20 | FJ Last_Twen | 18,2 | 3,52 |
| 21 | FJ Last_Twen2 | 18,4 | 3,35 |
| 22 | Boris Konrad 1 | 23,2 | 0,98 |
| 23 | Boris Konrad 2 | 23,5 | 1,12 |
| 24 | LateTFT | 23,5 | 1,12 |
| 25 | StrategyByBas (2) | 26,3 | 3,23 |
| 26 | Mugging Through | 26,4 | 2,06 |
| 27 | Mugging Through (2) | 26,4 | 1,28 |
| 28 | StrategyByBas | 26,9 | 2,3 |
| 29 | Hard Tit-for-Tat | 27,7 | 1,19 |
| 30 | Slow Tit-for-Tat | 30,2 | 0,87 |
| 31 | Tit-for-two-tats | 30,7 | 1,27 |
| 32 | S-TFTT | 31,1 | 0,94 |
| 33 | Go by majority | 33,2 | 0,6 |
| 34 | Der Zorn Gottes (2) | 36 | 1 |
| 35 | Sauer2 | 36 | 1,18 |
| 36 | Sauer1 | 36,6 | 1,2 |
| 37 | w-Random | 36,6 | 2,5 |
| 38 | Der Zorn Gottes | 36,9 | 1,76 |
| 39 | w-Random (2) | 37,3 | 3,13 |
| 40 | Verzögerte Bestrafung | 40,1 | 0,7 |
| 41 | Zweiundvierzig | 40,8 | 1,08 |
| 42 | adamone | 41,4 | 0,8 |
| 43 | He - Man | 43,2 | 0,4 |
| 44 | Always cooperate | 44 | 0,63 |
| 45 | TBattle Version 4 | 45,6 | 0,66 |
| 46 | Pavlov | 46,2 | 0,98 |

wird fortgesetzt. . .

... Fortsetzung von der letzten Seite

| Platz | Strategie Name | Mittelwert über Plätze | Standardabweichung über Plätze |
|-------|-----------------------|------------------------|--------------------------------|
| 47 | TBattle Version 4 (2) | 46,3 | 1,27 |
| 48 | evaWobeGa (2) | 48,8 | 1,6 |
| 49 | Spite | 48,9 | 0,83 |
| 50 | evaWobeGa | 49,9 | 1,14 |
| 51 | adamtwo | 50,2 | 0,87 |
| 52 | Young Capone | 52,1 | 0,54 |
| 53 | Green Shark | 52,8 | 0,4 |
| 54 | Raskolnikoff | 54,3 | 0,64 |
| 55 | Mistrust | 54,9 | 0,54 |
| 56 | Hard majority | 55,8 | 0,4 |
| 57 | MN2 | 57,5 | 0,5 |
| 58 | Major Eisberg | 57,7 | 0,78 |
| 59 | Throat | 58,8 | 0,4 |
| 60 | Dirk Alpha | 60 | 0 |
| 61 | Prober | 61,3 | 0,46 |
| 62 | Gartmann2 | 61,7 | 0,46 |
| 63 | Gralla 50 | 63,8 | 0,75 |
| 64 | Gralla 50 (2) | 63,9 | 0,94 |
| 65 | Per kind | 64,5 | 0,81 |
| 66 | Tester | 65,8 | 0,6 |
| 67 | Dirk Beta | 67 | 0 |
| 68 | Random | 68,1 | 0,3 |
| 69 | Joss | 68,9 | 0,3 |
| 70 | Gartmann1 | 70 | 0 |
| 71 | Per nasty | 71 | 0 |
| 72 | Resistance is futile | 72 | 0 |
| 73 | Dreiundzwanzig | 73 | 0 |
| 74 | Tasty Fish | 74,1 | 0,3 |
| 75 | Grayskull | 75,1 | 0,3 |
| 76 | Minor Eisberg | 76,5 | 0,67 |
| 77 | Betonschuh | 77,1 | 0,54 |
| 78 | Rasumichin | 78,4 | 0,49 |
| 79 | Sklave | 78,7 | 2,41 |
| 80 | Deep | 79,4 | 0,49 |
| 81 | Always defect | 80,7 | 0,46 |
| 82 | Delling | 82 | 0 |

7.3.4. Erkennungssequenzen

Es existiert die Möglichkeit, im Turnierverlauf selbst zu erkennen, welche Strategien Master bzw. Slave sind. Wegen variabler Erkennungssequenzen und nicht-deterministischem Verhalten können diese jedoch kaum von anderen Strategien unterschieden werden. So müssen z. B. Erkennungssequenzen nicht zusammenhängend sein (z. B. Erkennungskooperationen/-defektionen auf Position 5,13,31,480,...)

und können auch zufallsabhängig sein. Hierbei ist es denkbar, dass viele verschiedene Erkennungssequenzen mit einer bestimmten Wahrscheinlichkeit gespielt werden. Der zugehörige Master bzw. Slave kann dann einfach alle möglichen gespielten Erkennungssequenzen erkennen, denn Berechnungszeit ist derzeit kein limitierender Faktor in IPD-Turnieren. Umgekehrt erkennt der Slave mit demselben Prinzip auch den Master. Bei solch einer Vorgehensweise gibt es zu der Spielrundenanzahl exponentiell viele potentielle Erkennungssequenzen und ein unabhängiger „Schiedsrichter“ hat hier keine Möglichkeit mehr, eine Kooperation an einer speziellen Erkennungssequenz zu erkennen.

Ein Beispiel einer realen Erkennungssequenz des Siegers im PG474 *Round-Robin* Turnier von 2005 findet sich in Tabelle 7.4.

Tabelle 7.4.: Die Erkennungssequenz des siegreichen Masters und seines Slaves im IPD-Turnier der PG474 in 2005. *C* bedeutet Kooperation und *D* Defektion. Es ist ersichtlich, dass vom Master sehr wenige Defektionen gespielt werden und die Länge der Sequenz ziemlich kurz gehalten ist. Das führt zu einer größeren Tendenz zur Kooperation mit anderen Strategien, was in der Regel besser bewertet wird. Im Gegensatz dazu verhält sich der Slave eher unkooperierend mit vielen Defektionen.

| <i>L</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------------------|---|---|---|---|---|---|---|---|---|---|
| Master (<i>P</i>) | C | D | C | C | D | C | | | | |
| Slave (<i>P</i>) | D | D | C | D | C | D | D | D | C | C |

7.3.5. Analyse der *Jerusalem over all*-Umgebung

In Tabelle 7.3 und in dem Varianzdiagramm 7.4 ist ersichtlich, dass sowohl die Varianz der Punkte als auch der Plätze über die zehn Runden in fast allen Fällen sehr klein ist. Das wird bestätigt durch Abbildung 7.5, in der die Punkte der Strategien über die einzelnen Durchgänge dargestellt sind. Die Punktzahl ist hierbei weitgehend unabhängig vom Durchgang.

So ist zum Beispiel die Standardabweichung der Plätze in über 80% der Fälle unter einem Platz. Die durchschnittliche Standardabweichung aller Strategien beträgt nur 0,717 Plätze. Es gibt einen Ausreißer namens *THModGradual2* (siehe Bild 7.6), der eine sehr hohe Standardabweichung von 11,638 aufweist.

Werden jedoch die zugehörigen Punkte in Tabelle 7.7 betrachtet, fällt auf, dass sich hier diese Strategie völlig normal verhält und sogar eine kleinere Standardabweichung hat.

Das wird bestätigt durch Abbildung 7.8, in der die Standardabweichung über die Punkte dargestellt wird: *THModGradual2* ist hier weitgehend unauffällig. Das ist möglich, da in dem Siegerfeld alle Strategien sehr dicht gedrängt sind und eine kleine Punktabweichung schon große Platzunterschiede zur Folge hat. Hierzu kann auch Abbildung 7.9 betrachtet werden, in der die Punktdifferenzen der Strategien zu ihren Nachfolgern abgebildet sind. Ersichtlich ist, dass die Punktdifferenzen bei den ersten 41 hinterher gemeinsam kooperierenden Strategien sehr gering ist.

Bei genauerer Betrachtung kommt die hohe Varianz von *THModGradual2* über die Plätze von einem einzigen abweichenden Durchlauf. Der Grund ist, dass hier trotz kleiner Punktveränderung nicht mehr die kooperierende Strategiemenge bis Platz 41 erreicht wird, die Strategie landet nämlich direkt dahinter auf dem Platz 42.

mittlere Plätze im Vergleich zu mittleren Punkten

| Strategie | mittl. Platz / Std | | mittl. Punkte / Std | |
|------------------------------------|--------------------|--------|---------------------|---------|
| Netzer | 1,0 | 0,000 | 13.255.152 | 827.542 |
| THModGradual1 | 2,3 | 0,458 | 13.239.660 | 824.730 |
| Simon2 | 3,6 | 0,490 | 13.208.973 | 828.171 |
| MN1 | 4,7 | 0,458 | 13.189.291 | 823.849 |
| Sebrassi IPD II | 6,1 | 0,539 | 13.182.381 | 822.751 |
| Simon | 6,4 | 0,917 | 13.180.241 | 825.544 |
| Schwalli Faku | 8,3 | 1,005 | 13.137.341 | 816.989 |
| Gradual | 9,7 | 0,900 | 13.130.658 | 821.774 |
| SchwallisGradual | 9,7 | 1,005 | 13.130.441 | 823.324 |
| Sebrassi IPD I | 9,9 | 0,943 | 13.129.681 | 820.253 |
| Meister | 11,9 | 0,300 | 13.121.809 | 818.956 |
| TitForTat Plus1 | 12,9 | 0,300 | 13.118.896 | 819.725 |
| THModGradual2 | 7,4 | 11,638 | 13.109.753 | 697.622 |
| S-TFT | 16,0 | 1,549 | 13.106.696 | 818.729 |
| Tit-for-tat | 16,3 | 1,418 | 13.106.659 | 818.910 |
| Nils-TFT | 16,7 | 1,847 | 13.106.571 | 818.567 |
| Nils-TFT (2) | 16,5 | 0,922 | 13.106.554 | 818.720 |
| Tit for two tats | 17,2 | 1,536 | 13.106.539 | 818.805 |
| Späte Rache | 16,3 | 2,369 | 13.106.375 | 819.052 |
| FJ Last_Twen Tit_for_tat | 19,3 | 2,100 | 13.095.137 | 822.749 |
| FJ Last_Twen Tit_for_tat with Rec. | 20,9 | 0,300 | 13.076.768 | 818.287 |
| Boris Konrad 1 | 21,9 | 0,300 | 13.026.292 | 813.691 |
| Boris Konrad 2 | 22,9 | 0,300 | 13.019.550 | 813.125 |
| StrategyByBas (2) | 24,4 | 0,663 | 12.999.903 | 807.679 |
| StrategyByBas | 24,4 | 0,490 | 12.996.051 | 808.782 |
| Mugging Through (2) | 26,4 | 0,663 | 12.966.661 | 811.177 |
| Mugging Through | 26,5 | 0,671 | 12.966.538 | 807.081 |
| LateTFT | 27,8 | 0,400 | 12.953.645 | 809.798 |
| Slow Tit-for-Tat | 29,3 | 0,781 | 12.945.439 | 806.756 |
| S-TFTT | 30,0 | 0,775 | 12.942.247 | 806.449 |
| Tit-for-two-tats | 30,4 | 0,663 | 12.942.087 | 807.152 |
| Hard Tit-for-Tat | 31,9 | 0,300 | 12.920.064 | 807.758 |
| Go by majority | 33,9 | 0,943 | 12.788.581 | 797.960 |
| w-Random (2) | 33,9 | 0,831 | 12.787.786 | 785.475 |
| w-Random | 33,9 | 0,831 | 12.782.158 | 791.963 |

Abbildung 7.3.: *Jerusalem*-Umgebung: Die Standardabweichung der Plätze ist bei über 80% der Fälle kleiner als eins. Die durchschnittliche Standardabweichung aller Strategien beträgt nur 0,717 Plätze. Dargestellt sind hier die erste Hälfte der 82 Strategien.

gemittelte Plätze und Std

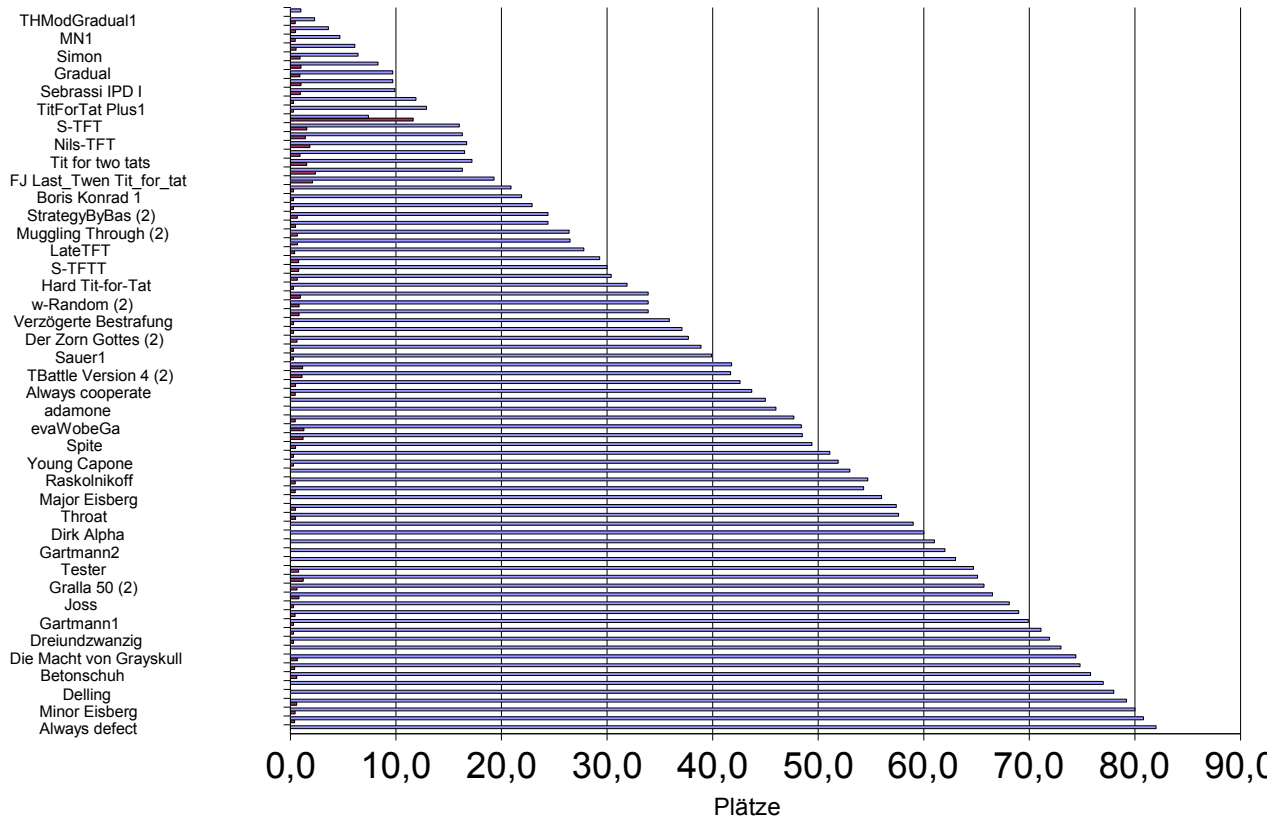


Abbildung 7.4.: Die Strategien sind hier nach Punktmittelwerten geordnet, deswegen ist die Angabe der Platzmittelwerte nicht vollständig monoton steigend. Die kleineren Balken stellen die Standardabweichung der Strategien dar. *THModGradual* ist die einzige Strategie, die eine relativ hohe Standardabweichung von über zehn Plätzen aufweist.

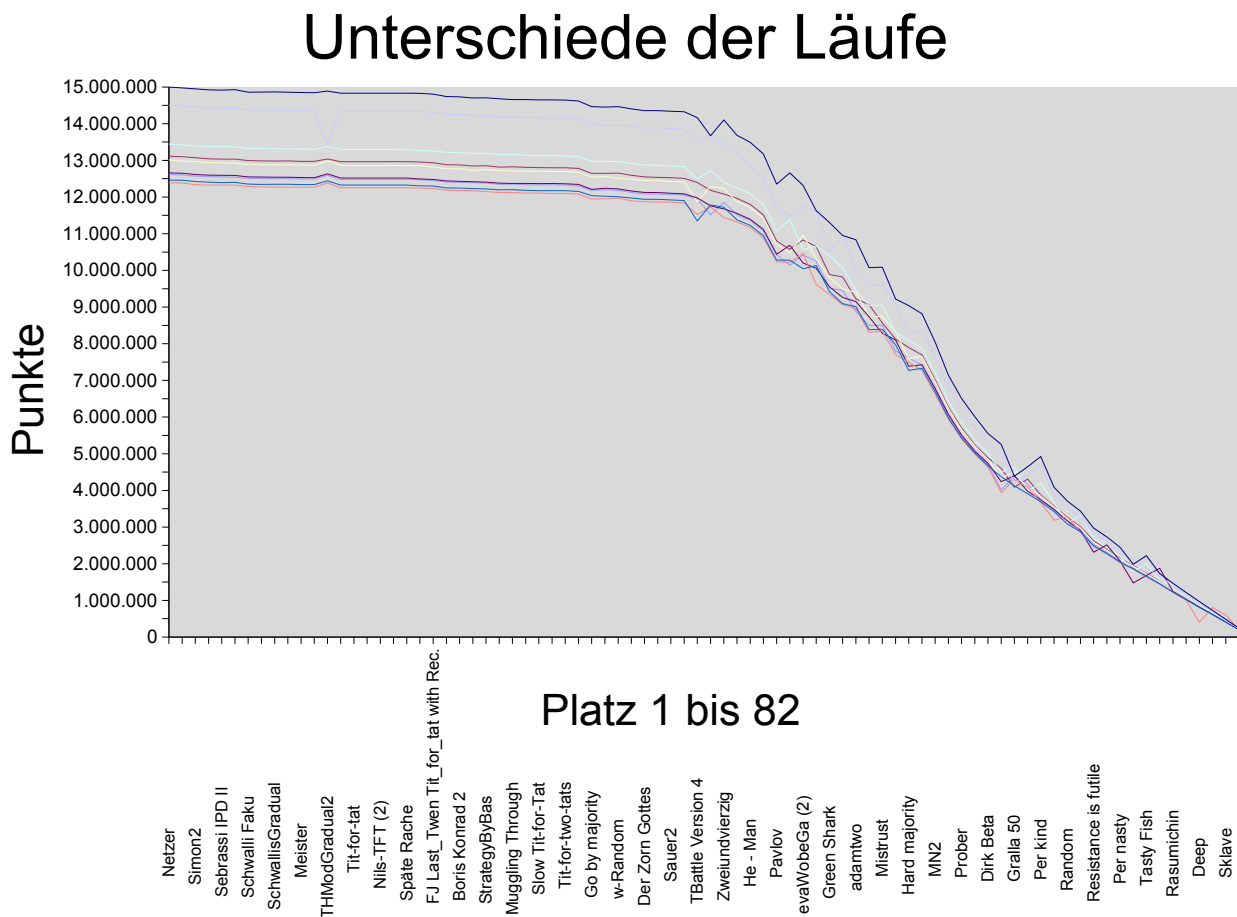


Abbildung 7.5.: Diese Abbildung zeigt das Verhalten der Punkte bei den einzelnen Läufen der zehn Iterationen. Das Verhalten der einzelnen Strategien hängt kaum von den unterschiedlichen Iterationen ab.

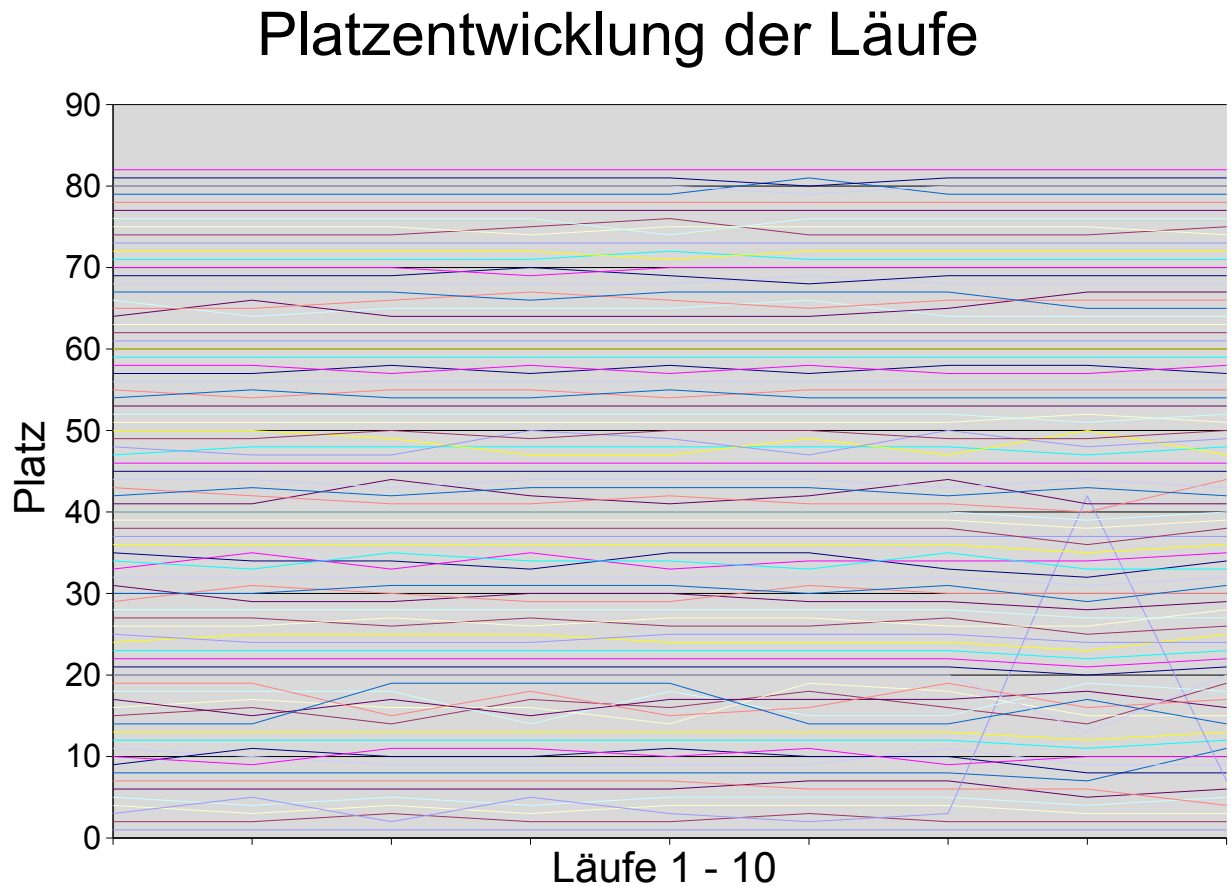


Abbildung 7.6.: Alle Strategien verhalten sich über die zehn Durchläufe annähernd gleich. Es gibt dennoch einen Ausreißer in Durchgang neun rechts unten im Bild. Diese Strategie namens *THModGradual2* sinkt dort von durchschnittlich Platz 7,5 auf den 42. Platz ab.

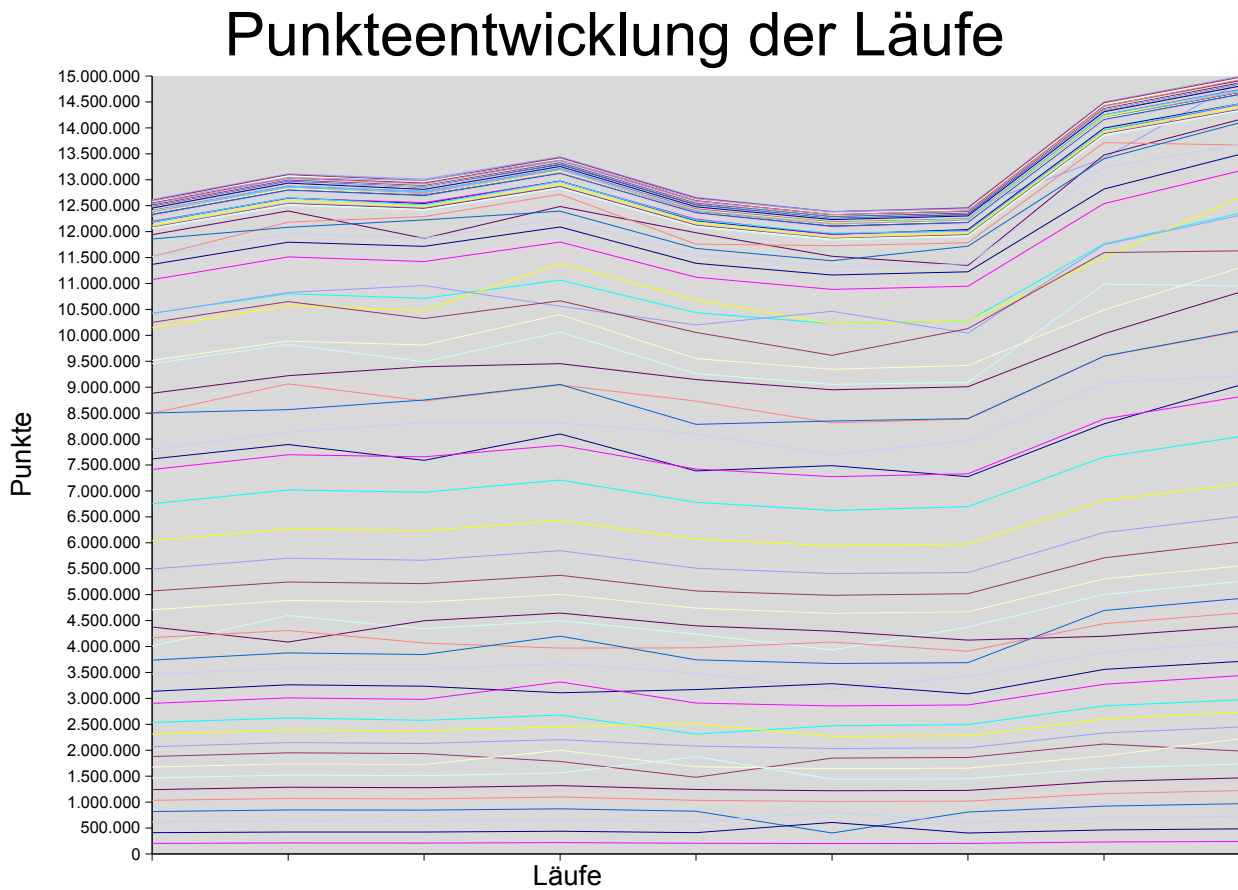


Abbildung 7.7.: Der Ausreißer *THModGradual2* ist allerdings in dieser Tabelle nicht mehr als solcher erkennbar. Das liegt daran, dass der Punkteunterschied im Durchgang neun relativ klein ist während die Plätze stark variieren. Insofern ist der vermeintliche Ausreißer keiner, der hohe Platzunterschied kommt dadurch, dass auf den ersten Plätzen alle Strategien sehr nahe beieinander liegen.

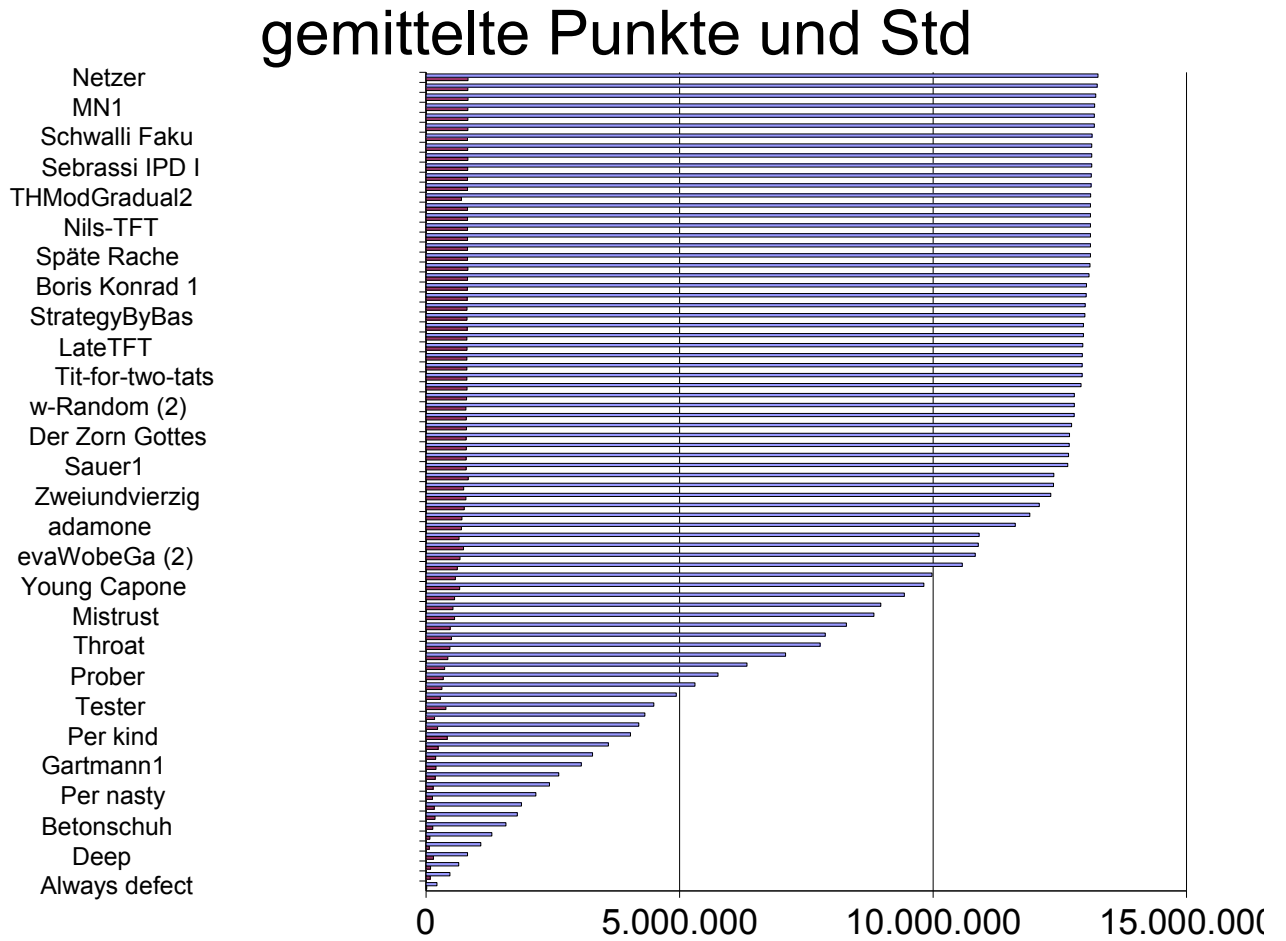


Abbildung 7.8.: Der dunkle Balken stellt die Varianz dar, der helle die Punktzahl. Hier ist *THMod-Gradual2* nicht auffällig, da die Standardabweichung nach den Punkten und nicht den Plätzen berechnet worden ist.

Punktdifferenzen

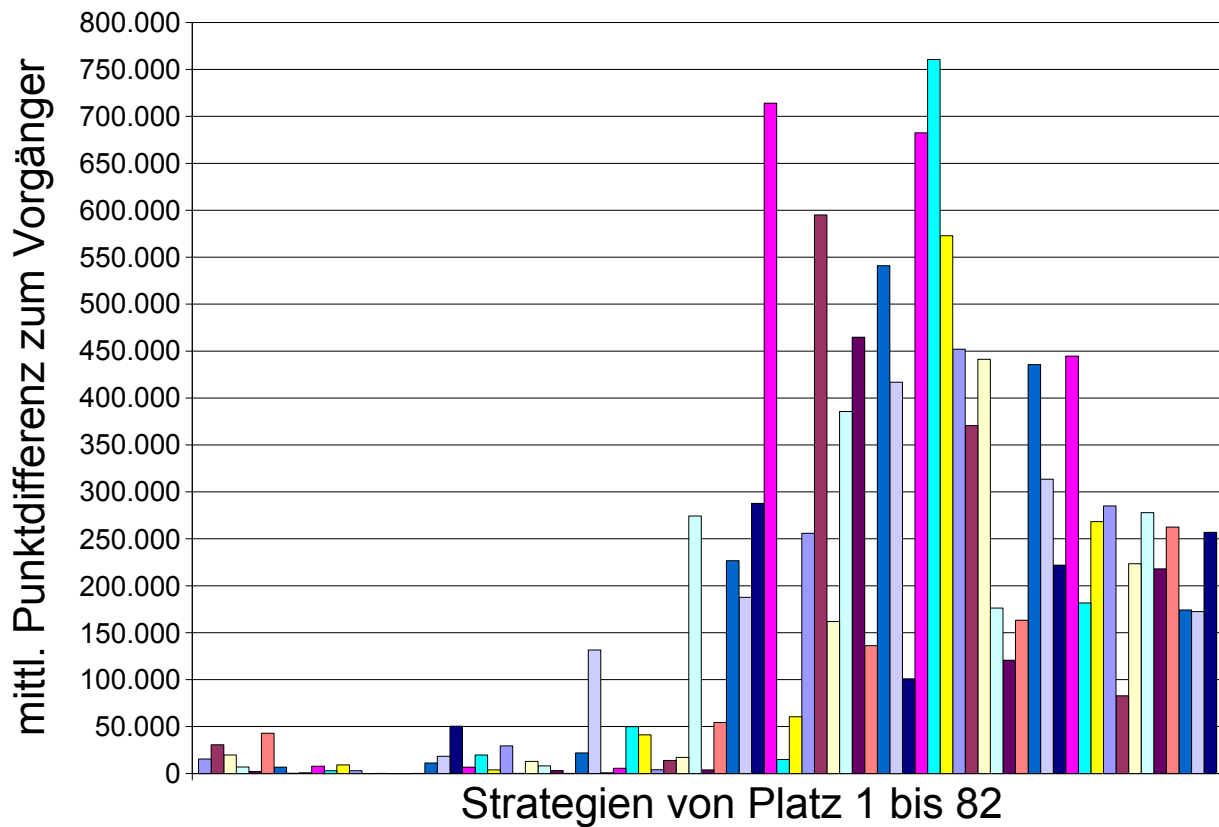


Abbildung 7.9.: Hier sind die Punktdifferenzen aller Strategien zu ihrem Nachfolger aufgeführt. Es ist ersichtlich, dass die ersten 41 Strategien eine vergleichsweise kleine Differenz haben. Das ist dadurch erklärbar, dass nach 41 entfernten Strategien von den insgesamt 82 nur noch miteinander kooperiert wird und es deswegen kaum Punktunterschiede gibt.

7.4. Zusammenfassung

Die Kleingruppe „Umgebungen“ konnte im zweiten Semester im Rahmen einer Projektgruppe an der Universität Dortmund einige interessante Ergebnisse erzielen. So wurde gezeigt, dass es für monotone Umgebungen nicht möglich ist das Master-Slave Problem komplett zu beseitigen. Obwohl der generelle Vorteil von Master-Slave-Strategien weiterhin besteht, ist es nun möglich, diesen enormen Vorteil mit Hilfe der neuen Umgebungen signifikant zu reduzieren. Die Definition der monotonen Umgebungen ist Beweis, dass Master-Slave-Strategien in normalen *Round-Robin*-Umgebungen immer einen Vorteil haben werden. Dennoch ist die genauere Unterscheidung oder Klassifizierung von Master-Slave-Strategien ein Schritt in die richtige Richtung, wenn es darum geht mit Master-Slave-Strategien umgehen zu können. Die neue Untersuchung von unkooperativen und kooperativen Slaves trägt so zu neuen Fragestellungen bei. Die Problematik mit Master-Slave-Strategien muss ggf. neu überdacht werden. Des Weiteren wurde das Problem mit Master-Slave-Strategien bisher fast ausschließlich mit nur wenigen Slave-Strategien untersucht. Wenn eine Vielzahl von Slave-Strategien für eine Master-Strategien zum Einsatz kommt, ist der Vorteil immer noch bei weitem zu groß. Hier könnten weitere Eliminationsmechanismen, wie z. B. in der ursprünglichen *Jerusalem*-Idee weiter verfolgt werden. Auch die Idee des Pre-Turniers könnte ausgeweitet werden, hier sind interessante Resultate zu erwarten. Mit der Einführung von Rauschen in die IPD-Umgebung sind wahrscheinlich weitere Auswirkungen auf Master-Slave-Strategien zu erwarten. Sind die Identifikationsmuster robust genug gegen die Fehler in der Datenübertragung? Und wie passen sich Master-Slave-Strategien an diese Umgebung an? Das Verlassen der monotonen Umgebung stellt eine schwierige Aufgabe da. Hier könnte der Master-Slave-Vorteil sehr stark reduziert werden, auch für eine Vielzahl von Slave Strategien. Die Frage bleibt wie weit hier der Geist des ursprünglichen Gedanken des PD erhalten bleibt.

8. Simulationen durch Softwareagenten

8.1. Einleitung

Eine der drei Untergruppen, die sich während der Arbeit der PG474 im zweiten Semester gebildet haben, befasste sich mit einer agentenbasierten Modellierung des Gefangenendilemmas. Agenten sind in diesem Zusammenhang autonome Prozesse, die dazu verwendet werden sollen, ein reales System zu simulieren. Vorteile dieses Vorgehens sind u. a., dass eine natürlichere Sicht auf das nachgebildete System ermöglicht wird und auch komplexe Systeme durch i. A. „einfache“ Agenten simuliert werden können.

Axelrod hat in [AH03] und in [HA05] zusammen mit Hammond mehrere agentenbasierte Modelle zur Untersuchung der *Evolution der Kooperation* vorgestellt. In [AH03] hat Axelrod nachgewiesen, dass sich das Phänomen des Ethnozentrismus' mit vergleichsweise einfachen Agenten simulieren lässt. In [HA05] haben Hammond und Axelrod die kooperationsfördernde Wirkung von lokaler Interaktion und Reproduktion (u. a. kombiniert mit der Fähigkeit der Agenten zur Gruppenunterscheidung) untersucht.

Dieser Ansatz war vielversprechend, nach Ansicht der Mitglieder der Kleingruppe allerdings in bestimmten Bereichen zu eingeschränkt und die Ergebnisse schienen in gewisser Hinsicht durch die Vorgaben des Modells vorhersehbar. Dadurch motivierte sich das Vorgehen, Axelrods Ansatz in seinen Grundzügen zu übernehmen und leicht abzuwandeln bzw. durch zusätzliche Komponenten zu erweitern. Während der Arbeit an den Erweiterungen haben sich die Mitglieder der Gruppe in zwei Untergruppen aufgeteilt. Die Vorgehensweise in den beiden Untergruppen, die Beschreibungen der Erweiterungen und einige der Ergebnisse sind im Folgenden sowie in zwei Ausarbeitungen im Anhang dieses Dokumentes nachzulesen.

8.2. Axelrods Viscosity-Model

Hammond und Axelrod haben in [AH03] und in [HA05] mehrere agentenbasierte Modelle zur Untersuchung der *Evolution der Kooperation* vorgestellt. Darunter befand sich u. a. das so genannte *Viscosity-Model* (M_{visc}).

Im M_{visc} sind die Agenten entweder reine Altruisten, d. h. sie kooperieren immer, oder reine Egoisten, d. h. sie kooperieren niemals. Des Weiteren besitzt jeder Agent ein *Reproduktionspotenzial* (*potential to reproduce*, PTR). Die Simulation wird auf einem Gitter der Größe 50×50 durchgeführt. Das Gitter hat *wrap-around* Kanten und damit hat jede Zelle des Gitters genau vier direkte Nachbarzellen.

Eine Iteration der Simulation besteht aus vier Schritten:

- Immigration

Ein Agent mit zufälliger Eigenschaft (entweder Altruist oder Egoist) wird in eine zufällig ausgewählte freie Zelle des Spielfeldes platziert.

- Interaktion

Am Anfang jeder Runde wird das PTR jedes Agenten auf 12% gesetzt. Jeder Agent spielt nun mit jedem seiner direkten Nachbarn, sofern diese existieren, ein PD bzw. ein *Public Goods Game*

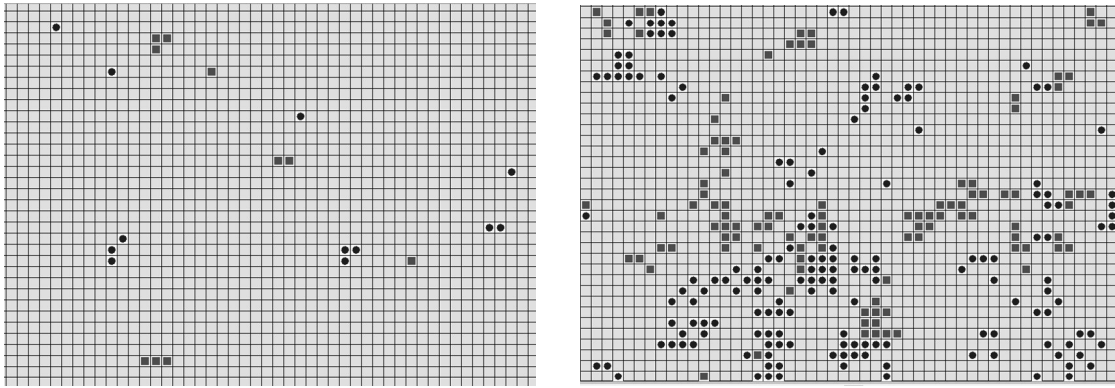


Abbildung 8.1.: Ein typischer Simulationsdurchlauf des M_{visc} nach 35 (links) und nach 350 (rechts) Iterationen. In dieser vergleichsweise frühen Phase der Simulation bilden die anfangs auf dem Spielfeld verstreuten Agenten Regionen mit einer homogener Strategie. Der schwarze Kreis repräsentiert einen Altruisten, das graue Quadrat repräsentiert einen Egoisten.

(*PGG*) mit zwei Spielern. Ein altruistischer Agent „hilft“ seinem Gegenüber, ein egoistischer Agent hilft seinem Gegenüber nicht. Hilfe zu geben reduziert das eigene PTR um $C = 1\%$. Hilfe zu empfangen erhöht das eigene PTR um $B = 3\%$.

- Reproduktion

Die Agenten auf dem Spielfeld werden in zufälliger Reihenfolge ausgewählt. Jedem Agenten wird die Chance gegeben, sich mit einer Wahrscheinlichkeit, anhand seines eigenen PTR, zu reproduzieren. Die Eigenschaften eines Agenten werden an seinen Nachkommen mit einer Mutationswahrscheinlichkeit von 0.5% weiter gegeben. Die Nachkommen werden in einer, soweit vorhanden, freien Nachbarzelle des Agenten platziert.

- Tod

Jeder Agent stirbt am Ende eine Runde mit einer Wahrscheinlichkeit von 10%.

8.3. Beschreibung der entwickelten Software

Für die nötigen Experimente war es notwendig, eine Software zu entwickeln. Da das M_{visc} ein nicht sehr komplexes Modell ist, wurde die Software in kürzester Zeit fertig gestellt. Zuerst wurde das Grundmodell entwickelt. Zu diesem konnten später durch kleine Veränderungen schnell die nötigen Erweiterungen programmiert werden. Die Software besteht aus vier Klassen, diese werden nachfolgend beschrieben. Als Programmiersprache wurde Java verwendet, da die Sprache allen PG-Teilnehmern bekannt war.

- `Agent.class`

Der Agent wird in der gleichnamigen Klasse beschrieben. Sie enthält die nötigen Parameter für das Handeln des Agenten. Das Handeln eines Agenten definiert die Strategie, die dieser Agent gegen seinen Gegner spielt. Diese Strategie darf nicht zu kompliziert sein, denn dies würde das Einfachheitsprinzip des Systems verletzen. Deshalb kann dieser Parameter nur zwei Werte annehmen, Kooperation oder Defektion. Im Modell für die Reputation sind die Strategien

nicht mehr rein Kooperation oder Defektion, sondern Reputation und Meinung. Diese Werte sind nicht mehr Boolean- sondern Doublewerte. Zur Vereinfachung der Analyse wurden diese Werte auf eine Dezimalstelle nach dem Komma gerundet und können zwischen 0 und 1 liegen, somit sind höchstens elf verschiedene Werte möglich. Im Punishment-Modell erhielt der Agent zusätzlich zur normalen Strategie (Kooperation oder Defektion) eine Punishment-Strategie, die als Feld implementiert wurde. Der Index des Feldes bezieht sich dabei auf die Punishment-Stage (Stufe der Bestrafung).

Die Strategie ist nur einer der Parameter, die diese Klasse enthält. Sie enthält auch die Farbe bzw. die Gruppenzugehörigkeit (nur bei dem Reputations-Modell), PTR und die Positionskordinaten auf dem Gitter. Weiterhin werden dort die nötigen Methoden für den Zugriff auf die einzelnen Parameter (get- und set-Methoden) sowie für die Erschaffung eines neuen Agenten beschrieben. Bei der Erzeugung eines Agenten werden seine Eigenschaften zufällig gewählt. Bei der Erzeugung eines Nachkommen erhält dieser die Eigenschaften seines Elters, wobei eine Mutationswahrscheinlichkeit berücksichtigt wird.

- **Grid.class**

Die Klasse Grid enthält, abgesehen von der Datenstruktur zur Speicherung der Agenten, die Kontrollmechanismen für das Verwalten des Grids. So kann durch einen Methodenaufruf überprüft werden, ob ein Feld schon belegt ist oder nicht. Um einen Agenten in eine Zelle zu platzieren, muss lediglich ein neuer Agent als Parameter an die geeignete Methode übergeben werden. Wenn ein Agent nicht platziert werden kann, weil z. B. bei der Reproduktion schon alle Nachbarfelder besetzt sind, werden die Agenten vernichtet, da kein freies Feld verfügbar ist.

- **Display.class**

Die Klasse Display hat die Aufgabe, die Simulation zu visualisieren. Nach jeder Generation wird das Fenster aktualisiert. Auf diese Weise ist es möglich, die Interaktion der Agenten mitzuverfolgen.

- **Simulation.class**

Die Aufgabe der Klasse Simulation ist die Durchführung der eigentlichen Simulation. Sie ruft die einzelnen Phasen, d. h. Immigration, Interaktion, Reproduktion und Tod, auf. Sie erschafft Agenten bei Immigration und übergibt diese an das Grid-Objekt. Bei Interaktion iteriert sie über das ganze Grid und lässt die Agenten gegen ihre Nachbarn antreten. Dabei wird nur überprüft, ob der Agent kooperiert, wenn dies der Fall ist, wird sein PTR verringert und das des Gegenübers erhöht. Es muss nun nicht überprüft werden, ob der Gegner ebenfalls kooperiert, denn nachdem der aktuelle Agent mit seinem Zug fertig ist, kommt ja irgendwann der Gegner des Agenten als aktueller Agent ins Spiel. Dabei ist der vorherige Agent sein Nachbar. Die Überprüfung des Agenten nach seiner Strategie ist bei dem Reputationsmodell fast gleich, jedoch wird nicht mehr überprüft, ob der Agent kooperiert, sondern es wird überprüft, ob der gegenüberliegende Agent genügend Reputation besitzt, um mit ihm zu kooperieren. Dazu wird der Parameterwert Meinung des Agenten mit dem Reputationswert des Gegenübers verglichen. Übersteigt die Reputation die Meinung, wird kooperiert. Bei dem Punishment-Modell wird zusätzlich zur normalen Strategie die Punishment-Strategie bei der Interaktion zweier Agenten gespielt. Nach der Interaktion werden, abhängig von dem Reproduktionspotenzial, Nachkommen erzeugt und an das Grid-Objekt zur Platzierung weitergegeben. Am Ende einer Generation sterben die Agenten mit einer bestimmten Wahrscheinlichkeit. Falls dies geschieht, wird auf dem Grid die entsprechende Stelle wieder freigegeben. Diese Klasse sorgt auch dafür, dass das Display-Objekt nach jeder Generation aktualisiert wird.

- **Data.class**

Diese Klasse schreibt die nötigen Simulationsergebnisse in verschiedenen Dateien um damit die spätere Analyse der Simulation zu ermöglichen.

8.4. Zusammenfassung „High Order Punishment“

8.4.1. Motivation

Altruistische Kooperation zwischen Individuen ist in der Natur weit verbreitet. Unter altruistischem Handeln wird allgemein selbstloses Handeln verstanden. Im Kontext des PD und des PGG bedeutet Altruismus, dass bei einseitiger Kooperation der Kooperierende einen Nachteil und nur der Nicht-Kooperierende einen Vorteil hat. Aus diesem Grund sollte eigentlich zwischen rational und egoistisch handelnden Individuen keine Kooperation entstehen.

Dieser Umstand führt direkt zu dem so genannten *Free-Rider-Problem*. Das Free-Rider-Problem tritt dann auf, wenn es unabhängig von der Reaktion des Gegenüber und der Gesellschaft von Vorteil ist, nicht zu kooperieren. Ein Individuum, welches mehr als den ihm zustehenden Anteil an einer begrenzten Resource konsumiert, wird als *Free-Rider* bezeichnet.

Es wurden bereits viele Faktoren und Mechanismen untersucht, die bei der *Evolution der Kooperation* förderlich sein können. Es kam die Frage auf, wie das Phänomen der Free-Rider verhindert oder zumindest eingeschränkt werden kann. Ein Faktor, der unter bestimmten Umständen Kooperation fördern kann, ist beispielsweise *Spatial structure* [NBM94, NS00, DH05]. Des Weiteren ist *direkte Reziprozität* ein kooperationsfördernder Faktor [DH05]. Mit direkter Reziprozität werden die Auswirkungen kontinuierlicher Interaktionen zwischen Individuen (wie z. B. beim IPD) bezeichnet. Bei der *direkten Reziprozität* wirkt der Aufbau einer Reputation kooperationsfördernd [SHN01, DH05]. Diese Reputation der jeweiligen Individuen kann u. a. auch zu Aktionen der Bestrafung und Belohnung führen.

Im Paper *High-Order Punishment and the Evolution of Cooperation* wird untersucht, wie sich die Möglichkeit zur Bestrafung auf das M_{visc} Modell von Hammond und Axelrod [HA05] auswirkt. In diesem Kapitel finden Sie eine kurze Beschreibung des erweiterten Simulationsmodells (das *Punishment-Model*, M_{pun}) und eine kurze Analyse der Simulationsergebnisse. Der komplette Artikel befindet sich im Anhang B.

8.4.2. Erweitertes Simulationsmodell

Im M_{visc} Modell von Hammond und Axelrod [HA05] sind die Agenten entweder Altruisten oder Egoisten (siehe auch Kapitel 8.2). Im M_{pun} Modell haben die Agenten eine zusätzliche Handlungsoption: die *Punishment Strategie*. Nach jeder Runde werden die Agenten darüber informiert, welche Strategie ihr direkter Nachbar gespielt hat. Wenn es sich beim Gegenüber um einen Egoisten handelt, haben die Agenten die Möglichkeit, diesen zu bestrafen. Jemanden zu bestrafen bedeutet, dass ein Anteil vom eigenen PTR aufgewendet wird um das PTR des Gegenüber zu reduzieren.

Mit dem M_{pun} Modell soll zuerst untersucht werden, ob diese Bestrafungsoption die Evolution der Kooperation im M_{pun} Modell fördert bzw. ob das Phänomen der Free-Rider im M_{pun} Modell verhindert oder zumindest eingeschränkt werden kann.

Bestrafung verursacht auf der einen Seite Kosten beim Bestraften, auf der anderen Seite allerdings auch Kosten beim Bestrafenden. Somit kann der Akt der Bestrafung auch als altruistisch angesehen werden. Im Zusammenhang mit der Möglichkeit zur Kooperation stellte sich die Frage: „Warum soll ich überhaupt kooperieren?“ – das *Dilemma der ersten Ebene*. Im Zusammenhang mit der Möglichkeit zur Bestrafung stellt sich nun die Frage: „Warum soll ich überhaupt bestrafen?“ – das *Dilemma der zweiten Ebene*. Das Free-Rider-Problem der ersten Ebene wurde somit durch ein Free-Rider-Problem der zweiten Ebene ersetzt. Dieser Prozess, in dem ein Dilemma einer Ebene durch ein Dilemma der nächsthöheren Ebene ersetzt wird, lässt sich unendlich oft fortsetzen.

Mit dem M_{pun} Modell soll des Weiteren untersucht werden, wie sich die Möglichkeit zur Bestrafung von Free-Rider der zweiten und dritten Ebene auf die Evolution der Kooperation im M_{pun} Modell auswirkt.

8.4.3. Durchführung der Simulation

Die Simulation wurde mit einer Variante der in Kapitel 8.3 vorgestellten Software durchgeführt. Ein einzelner Simulationsdurchlauf bestand aus 2000 Runden und jeder einzelne Durchlauf wurde insgesamt zehn mal wiederholt. Die initialen Startparameter der Simulation waren weitestgehend identisch mit den Standardwerten der Simulation von Hammond und Axelrod [HA05]. Hammond und Axelrod haben die Auswirkung verschiedener Kosten-Nutzen-Verhältnisse (*cost-benefit ratio*, CBR) auf ihr M_{visc} Modell untersucht [HA05]. Das CBR ist das Verhältnis zwischen den Kosten C , Hilfe zu geben und dem Nutzen B , Hilfe zu empfangen. Um die Ergebnisse des M_{visc} Modell und des M_{pun} Modell vergleichen zu können, wurden auch beim M_{pun} Modell verschiedene Kosten-Nutzen-Verhältnisse untersucht. Der Wert des Nutzens B war bei allen Simulationsdurchläufen fest 3%, die Kosten C durchliefen das Intervall [0%, 3%].

Des Weiteren wurde die Auswirkung verschiedener Kosten-Strafe-Verhältnisse (*cost-fine ratio*, CFR) auf das M_{pun} Modell untersucht. Das CFR ist das Verhältnis zwischen den Kosten P , jemanden zu bestrafen und den Kosten F , bestraft zu werden. Es wurden mehrere Simulationsdurchläufe für einen festen Wert $F = 1\%$ durchgeführt. Bei diesen Simulationsdurchläufen durchliefen die Kosten P das Intervall [0%, 1%]. Weiterhin wurden mehrere Simulationsdurchläufe für einen festen Wert $F = 3\%$ durchgeführt. Bei diesen Simulationsdurchläufen durchliefen die Kosten P das das Intervall [0%, 3%].

Um das Free-Rider-Problem auch auf der höheren Ebene zu untersuchen, wurden die einzelnen Simulationsdurchläufe mit unterschiedlichen *Punishment-Level* θ durchgeführt. Ein $\theta = 0$ bedeutet, dass die Agenten keinerlei Möglichkeit haben, andere Egoisten zu bestrafen. Ein $\theta = 1$ bedeutet, dass die Agenten die Möglichkeit haben, andere Egoisten (Free-Rider der ersten Ebene) zu bestrafen. Wenn das $\theta = 2$ ist, können die Agenten Free-Rider der ersten und der zweiten Ebene bestrafen. Wenn das $\theta = 3$ ist, können die Agenten Free-Rider der ersten, der zweiten und der dritten Ebene bestrafen.

8.4.4. Ergebnisse und Analyse

In diesem Kapitel finden Sie eine kurze Zusammenfassung der Simulationsergebnisse. Alle Ergebnisse und eine detailliertere Analyse finden Sie im Anhang B oder im Internet unter [B⁺06].

- Im Vergleich zum M_{visc} Modell (*Punishment-Level* $\theta = 0$) kann im M_{pun} Modell (*Punishment-Level* $\theta = \{1, 2, 3\}$) Kooperation auch in einer unfreundlicheren Umgebung entstehen.¹
- Je höher das *Punishment-Level* θ ist, desto mehr wird Kooperation in einer unfreundlicheren Umgebung gefördert. In jedem der sechs Diagramme der Abbildung 8.2 bedeutet ein höheres *Punishment-Level* θ , dass der Kurvenverlauf ein wenig weiter nach rechts in eine unfreundlichere Umgebung rückt.
- Je höher die Auswirkungen F der Bestrafung sind, desto mehr wird Kooperation in einer unfreundlicheren Umgebung gefördert. In der linken Spalte der Abbildung 8.2 (die Diagramme *a*, *c* und *e* in der Abbildung 8.2) sind die Kosten vergleichsweise niedrig ($F = 1\%$). In der rechten

¹In einer „freundlichen“ Umgebung sind die Kosten für die Hilfe vergleichsweise niedrig (z. B. $C = 1\%$ und $B = 3\%$). In einer „unfreundlichen“ Umgebung sind die Kosten für die Hilfe dagegen vergleichsweise hoch (z. B. $C = 2\%$ und $B = 3\%$).

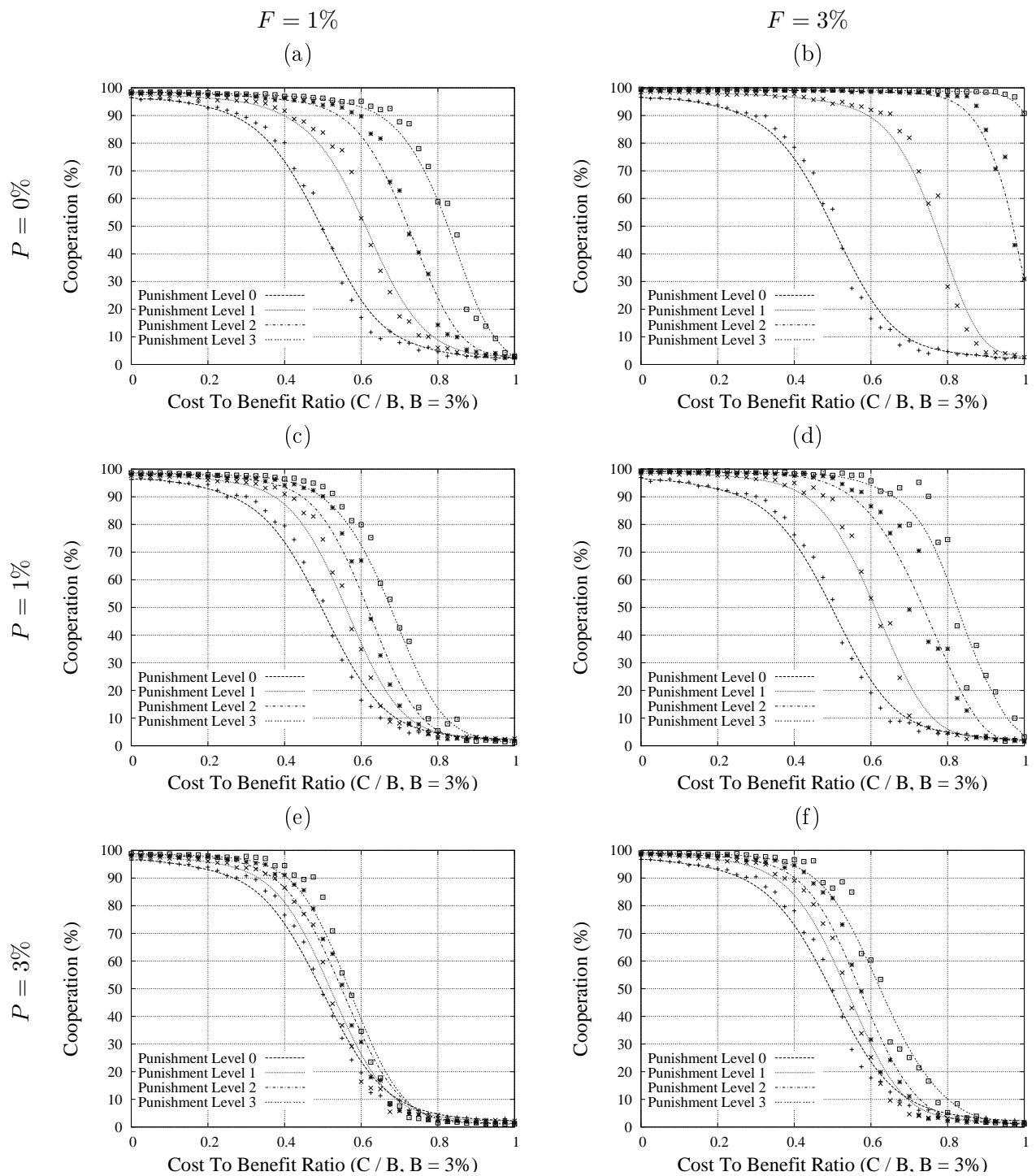


Abbildung 8.2.: Im Vergleich zum M_{visc} Modell kann im M_{pun} Modell Kooperation auch in einer unfreundlicheren Umgebung entstehen.

Spalte (die Diagramme b , d und f in der Abbildung 8.2) sind die Kosten vergleichsweise hoch ($F = 3\%$).

- Selbst wenn die Kosten für die Bestrafung P um ein vielfaches größer als die Auswirkung der Bestrafung F sind, ist das Kooperationsniveau im M_{pun} Modell ein wenig höher als im M_{visc} Modell (die Diagramme e und f in der Abbildung 8.2).

8.5. Zusammenfassung „The Impact of Group-Reputation“

Das Paper mit dem Titel “The Impact of Group-Reputation in a Multi-Agent Environment“ beschäftigte sich mit dem Einfluss der Reputation auf die Entwicklung der Kooperation in einer agentenbasierten Simulation. Das Paper erweiterte das von Robert Axelrod [AH03] eingeführte Modell sowohl um Gruppenreputation als auch um Reputation von Individuen. Zuerst wurde eine kurze Einführung in die Problematik des IPD gegeben und erläutert, weshalb es lohnenswert ist, sich mit Reputation zu beschäftigen. Anschließend wurden die verwendeten Modelle erläutert, sowie die Änderungen zu Axelrods Simulation herausgestellt. Die gefundenen Ergebnisse wurden vorgestellt und ausgewertet, sowie ein Fazit gezogen und dargelegt, welche Ergebnisse weiterer Untersuchungen bedürfen.

Die Gründe für die Wahl der Reputation waren vielfältiger Natur. Zum einen kann der Ruf einer Person, also das Ansehen, welches sie bei anderen genießt, entscheidend das Verhalten, welches ihr gegenüber an den Tag gelegt wird, beeinflussen. Das kann entweder positiv oder negativ beeinflusst werden. Auch in unserer Gesellschaft spielt der Ruf einer Person oder Institution eine entscheidende Rolle.

Zum anderen wurde durch die Einführung der Reputation die Situation in der Simulation für die Agenten verändert. Spielten die Agenten in Axelrods Originalsimulation noch jede Runde für sich ein PD-Spiel, wurde dies durch die Reputation dahingehend verändert, dass die Agenten nun vor eine Situation gestellt wurden, die eher dem IPD gleicht. Man war somit in der Lage, die Ergebnisse und Strategien, welche aus dem durchgeführten Turnier im letzten Jahr gewonnen wurden, in die Simulation teilweise einzubauen. Darüber hinaus herrschte die Meinung vor, dass eine dem IPD ähnelndere Situation für die Agenten realistischer wäre, bezogen auf Übertragbarkeit auf Situationen im realen Leben.

Wie bereits erwähnt, beruhte die in diesem Paper durchgeführte Simulation auf der von Axelrod. Die Simulation wurde in vier Phasen eingeteilt: *Immigration*, *Interaktion*, *Reproduktion*, *Tod*. In der *Immigrationsphase* wurden Agenten einer bestimmten Gruppe auf einem Spielfeld in Gitterform zufällig eingesetzt. Dieses Spielfeld wurde dermaßen konstruiert, dass jedes Feld, auch die Ecken, genau vier Nachbarfelder hatte. In der *Interaktionsphase* spielten alle benachbarten Agenten eine Runde des PD gegeneinander, in der die erreichten Punkte dieser Runde auf die Wahrscheinlichkeit für Nachwuchs (PTR) aufsummiert oder abgezogen wurden, je nachdem wie der Agent spielte. Die Entscheidung, welcher Zug gewählt wurde, wurde durch je eine Variable für die eigene Gruppe und für alle anderen Gruppen zusammen bestimmt. Diese PTR war eine Prozentangabe, nach der entschieden wurde, ob dieser Agent in der *Reproduktionsphase* einen Sprössling auf das Spielfeld setzen durfte. Dieser wurde zufällig auf ein freies Nachbarfeld positioniert, wobei die Eigenschaften des Kindes mit einer bestimmten Wahrscheinlichkeit mutiert wurden. Die PTR wurde reduziert, wenn der Agent selber kooperierte (der Wert der Kooperationskosten lag je nach Experiment zwischen null und drei) und erhöht, wenn mit ihm kooperiert wurde (immer um drei Punkte). In der *Todesphase* bestand schließlich für jeden Agenten, außer den Neugeborenen, eine feste Wahrscheinlichkeit zu sterben.

Die erste Erweiterung bestand in der Hinzunahme von Reputation. Diese wurde als der Anteil von Kooperationen an der Anzahl aller Züge eines Agenten definiert. Des Weiteren wurde die Farbzugehörigkeit entfernt, indem nur eine Gruppe implementiert wurde. Damit wurde der Fall simuliert, dass jeder Agent einer eigenen Gruppe angehörte. Eine notwendige Änderungen betraf die zu wählende Entscheidung, ob kooperiert oder defektiert werden sollte. Damit die Reputation überhaupt

einen Effekt hatte, wurden die alten Parameter, welche den Zug aufgrund der Gruppenzugehörigkeit des Gegners bestimmten, entfernt und eine neue Variable, die Erwartung, eingeführt. Nur wenn die Reputation des Gegners größer oder gleich dem eigenen Erwartungswert war, wurde kooperiert, ansonsten defektiert. Sowohl die Reputation als auch die Erwartung waren Gegenstand einer möglichen Mutation.

Die zweite Erweiterung bestand in der Einführung der Gruppenreputation. Zu diesem Zweck wurden wieder unterschiedliche Gruppen eingeführt und jede Gruppe mit einer eigenen Reputation ausgestattet, welche sich aus dem Mittelwert der Reputationen aller Agenten dieser Gruppe zusammensetzte. Da wieder verschiedene Gruppen existierten, musste auch die Erwartung angepasst werden. So hatte jeder Agent eine eigene Erwartung an jede einzelne Gruppe, auch an seine eigene.

Als Ergebnisse wurden vornehmlich die Abweichungen in Tabellenform angegeben, die zwischen dem Anteil der Kooperation von dem hier durchgeführten Modell zu dem von Axelrod durchgeführten festgestellt wurden. Dies geschah für jedes der beiden hier verwendeten Modelle durch den Durchschnitt über alle Runden und über die letzten 100 Runden (von 2000). Für jedes der beiden Modelle wurden Tabellen mit den gleichen Belegungen erstellt, damit auch eine Vergleichbarkeit zwischen den Modellen gegeben war. Die gezeigten Tabellen stellten selbstverständlich nur einen Teil der produzierten Ergebnisse dar. Um die Vergleichbarkeit mit Axelrods Ergebnissen sicherstellen zu können, wurden die gleichen Werte wie in seinem Modell ausgegeben. Untersucht wurden vor allem die Auswirkungen, welche Reputation an sich auf die Simulation hat. Ebenso wurde in beiden Modellen untersucht, was bei erhöhten Kosten bei Kooperation geschieht. In dem zweiten Modell wurde zusätzlich untersucht, welchen Einfluss die Anzahl der Gruppen hat. Darüber hinaus wurde anhand von Regressionsbäumen bestimmt, welche Faktoren welchen Einfluss auf die Ergebnisse hatten.

In der Analyse des ersten Modells wird vor allem herausgestellt, dass die Reputation einen erheblichen und positiven Einfluss auf die Entwicklung der Kooperation hat. Auch unter widrigen Anfangsbedingungen, wie hohe Kosten für die Kooperation oder hohe Erwartungen gepaart mit niedriger Reputation, wird immer ein höheres Level der Kooperation erreicht als bei Axelrod. Unter einigen Anfangsbedingungen war der Anteil der Kooperation im vorliegenden Modell durchschnittlich nicht so hoch wie bei Axelrod, doch auch in diesen Fällen lag der Anteil gegen Ende der Simulationen deutlich höher, was zeigte, dass auch der Anstieg des Anteils der Kooperation deutlich höher war als bei Axelrod. Es konnte also ein gewisses Maß der Robustheit der Reputation nachgewiesen werden. In der Analyse des zweiten Modells, Gruppenreputation, wurde gezeigt, dass die Effekte, welche im ersten Teil der Analyse erläutert wurden, auch im zweiten Teil Gültigkeit hatten, dass also Reputation einen durchweg positiven Effekt auf die Entwicklung der Kooperation hatte. Darüber hinaus wurde festgestellt, dass mit der eigenen Gruppe wesentlich weniger kooperiert wurde als ohne verschiedene Gruppen. Leider konnte für diese Beobachtung noch keine Erklärung gefunden werden, sie muss also Gegenstand weiterer Untersuchungen bleiben. Im dritten Teil der Analyse wurde anhand der Regressionsbäume untersucht, welche Faktoren aus welchem Grund einen Einfluss auf die Entwicklung der Kooperation hatten. Es wurde festgestellt, dass unter einer bestimmten Anfangsbelegung der Reputation (Maximalwert) und der Erwartung (Minimalwert) die erzielten Ergebnisse unabhängig von anderen Faktoren immer sehr hoch waren. In den anderen Fällen waren vor allem die Kosten entscheidend, welche für das Kooperieren mit anderen Agenten aufgewendet werden mussten.

Zusammenfassend lässt sich über die Reputation einzelner Agenten sagen, dass sie im Allgemeinen die Kooperation fördert, vor allem wird dies unter schwierigen Simulationsbedingungen deutlich. Auch zeigte sich, dass nicht nur das erreichte Level der Kooperation höher ist, sondern auch der Anstieg des Anteils der Kooperation mit Reputation deutlich höher ist. Die Gruppenreputation hingegen sorgte für ein geringeres Maß an Kooperation innerhalb der gleichen Gruppe, dafür für ein höheres Maß an Kooperation mit anderen Gruppen. Die Anzahl der verschiedenen Gruppen hingegen spielt keine Rolle für die Kooperation mit der eigenen Gruppe. Weiterhin stellte sich heraus, dass bei einem hohen Erwartungswert die Kosten für die Kooperation eine entscheidende Rolle spielen, wohingegen eine hohe Anfangsreputation aller Agenten nur unwesentlich von höheren Kosten für die Kooperation

beeinflusst wird.

8.6. Zusammenfassung

Zum Ende des zweiten Semesters der Projektgruppe wurden von der Untergruppe Softwareagenten Ergebnisse in zwei unterschiedlichen Ansätzen erarbeitet. Zum einen wurde von einem Teil der Kleingruppe die Auswirkung von Bestrafung auf die Evolution der Kooperation in einem agentenbasierten Modell betrachtet. Der andere Teil der Kleingruppe beschäftigte sich hingegen mit den Konsequenzen von Reputation. Am Ende der Zusammenfassung dieser Ergebnisse steht ein kurzes Fazit, welches die Kleingruppe über ihre Arbeit im zweiten Semester gezogen hat.

8.6.1. High-Order Punishment

Die Möglichkeit zur Bestrafung fördert Kooperation auch in unfreundlichen Umgebungen. Allerdings kann der untersuchte Bestrafungsmechanismus das Phänomen der Free-Rider nicht komplett unterbinden. Das betrachtete Modell zur Bestrafung ist jedoch dazu geeignet, die Mechanismen und Wechselwirkungen von Bestrafung und Kooperation zu untersuchen. Ein interessanter Aspekt für mögliche weitere Nachforschungen ist die Kombination von Bestrafungs- und Belohnungsmechanismen.

8.6.2. Impact of Group-Reputation

Die Einrichtung von Reputation unterstützt die Entwicklung von Kooperation. Dies wird besonders gut ersichtlich, wenn Umgebungen mit hohen Kosten für den Akt der Kooperation betrachtet werden. Bei den betrachteten Anfangskonfigurationen für Reputation und der Anforderung an den Ruf des Gegenübers wurde klar, dass sich ungünstige Belegungen erst sehr spät ausgleichen und die Evolution von Kooperation dadurch etwas gehemmt wird. Während Reputation in einer Umgebung mit mehreren Gruppen zu weniger Kooperation mit der eigenen Gruppe führt, wird die Kooperation mit anderen Gruppen sehr stark gefördert und damit ethnozentristisches Verhalten verringert. Die Anzahl der Gruppen spielt bei der Kooperation mit der eigenen Gruppe nie eine Rolle, kann aber durch bestimmte Einflüsse den Grad der Kooperation gegenüber anderen Gruppen verändern.

8.6.3. Softwareagenten und das IPD im Rahmen der Projektgruppe

Die Untergruppe, welche sich im zweiten Semester mit der Untersuchung von Softwareagenten und dem IPD beschäftigte, konnte keine Ergebnisse des Turniers vom Campusfest aus dem ersten Semester verwerten. Des Weiteren waren keine Ergebnisse der anderen Untergruppen verwendbar. Dies lag an der betrachteten Thematik. Softwareagenten sollen so einfach wie möglich konstruiert werden. Eine Erweiterung der Modelle um komplexere Strategien wie *TFT* oder *GRADUAL* hätte gegen die Natur der agentenbasierten Modellierung verstoßen. Durch diese Einschränkung kam es zu wenig Synergieeffekten. Eine zukünftige Projektgruppe sollte dies beachten und versuchen, mögliche Untergruppen inhaltlich aufeinander abzustimmen.

Allerdings bot das gewählte Thema genügend Möglichkeiten um neben theoretischen Erfahrungen, wie etwa der Analyse von Verhalten in komplexen Systemen, auch Praxis zu vermitteln. Schließlich mussten mehrere Simulationsumgebungen entworfen werden, die einem hohen Standard schon alleine deswegen genügen mussten, weil sie von den eingesandten Konferenzarbeiten referenziert wurden. Neben den gewonnenen Erfahrungen war die Aktualität des Themas sowie die realitätsnahe Anwendungsmöglichkeit ein motivierender Faktor bei der Arbeit.

9. Strategiegewinnung durch EA

9.1. Einleitung

9.1.1. EA

Der Gedanke, neue Strategien für das IPD mit Hilfe evolutionärer Algorithmen zu entwickeln, liegt recht nah. Der Prozess, in dem von Menschen kreierte Turnierstrategien entstehen, kann auch als evolutionär aufgefasst werden: Wenn ein Konzept bei einem Turnier erfolgreich ist, wird es von anderen Teilnehmern später (meist in leichter Variation) imitiert. Weniger erfolgreiche Konzepte hingegen sterben aus, d. h. werden nicht mehr benutzt.

Bereits Axelrod nutzte evolutionäre Algorithmen, um neue Turnierstrategien zu finden und um die Evolution der Kooperation zu untersuchen [Axe97]. Er und viele andere nutzen aber meistens vergleichsweise einfach strukturierte Repräsentationen, wie z. B. Look-Up-Tables, endliche Automaten und neuronale Netze. Das Ziel dieser Gruppe ist es, komplexere Repräsentationen zu untersuchen und mit diesen erfolgreiche Strategien zu generieren.

Hierbei sollen zwei verschiedene Repräsentationen in zwei verschiedenen Umgebungen (s. u.) untersucht werden. Die Frage der Repräsentation ist essentiell für den Suchprozess, da eine zu einfache Repräsentation gute Lösungen von vornherein ausschließen könnte.

9.1.2. Internes Turnier

Im zweiten Semester wurde regelmäßig ein internes Turnier durchgeführt, an dem alle Mitglieder der Projektgruppe teilnehmen konnten. Genutzt wurde dazu die im ersten Semester entwickelte Turniersoftware. Jederzeit konnten neue Strategien eingeschickt werden, eine Auswertung wurde mittels eines Turniers wöchentlich vorgenommen und veröffentlicht.

Dadurch wird, ähnlich wie im vorigen Absatz beschrieben, die natürliche Evolution simuliert. Es war zu erwarten, dass erfolgreiche Strategien imitiert und erfolglose nicht mehr eingeschickt werden. Der Sinn bestand einerseits in der (durchaus erfolgreichen) Anregung der kreativen Kräfte der PG-Teilnehmer durch die Konkurrenzsituation. Andererseits stellte das interne Turnier ein willkommenes Übungs- und Testfeld für die EA-Gruppe dar, welche ihre evolutionär entwickelten Strategien so leicht an denen der anderen PG-Teilnehmern testen konnten.

9.2. Beschreibung der Ansätze

9.2.1. LGP-Ansatz

Viele in der Praxis vorkommende Strategierepräsentationen schränken den Raum der darstellbaren Strategien sehr stark ein. Beispielsweise wurde in [BDM96] eine Strategie namens *GRADUAL* präsentiert, welche nicht mit den in Axelrods erster Untersuchung zur evolutionären Generierung von Strategien [Axe97] vorgeschlagenen Look-Up-Tables dargestellt werden kann. Auch andere Repräsentationen, wie z. B. endliche Automaten [Fog93] oder neuronale Netze [CY05] umfassen diese Strategie nicht.

Daher liegt es nahe, einen Ansatz zu verwenden, welcher die größtmögliche Darstellbarkeit von Strategien erlaubt. Aus diesem Grund wurde im Rahmen der Projektgruppenarbeit zur Evolution

von Strategien auf die Technik des linearen genetischen Programmierens [BNKF98] zurückgegriffen. Bei dieser besteht eine Strategie aus einer Liste von Befehlen, welche von einer simulierten 3-Registermaschine ausgeführt wird. Dieses Programm wird vor jedem Zug, welchen die Strategie macht, ausgeführt und das Ergebnis der Berechnung als Entscheidung, ob die so dargestellte Strategie nun kooperiert oder defektiert, verwendet.

Theoretisch können alle berechenbaren Strategien auf diese Weise dargestellt werden. Da aber sowohl die Rechenzeit als auch der Speicherplatz beschränkt sind, ist dies in der Praxis nicht möglich. Dennoch ist dies sicherlich einer der wenigen Ansätze, welcher wirklich alle sinnvollen Strategien, die in letzter Zeit vorgeschlagen wurden, abdeckt.

Die Menge der verfügbaren Befehle der 3-Registermaschine wird später beschrieben. Diese Befehle arbeiten auf einer Menge von insgesamt 15 Registern:

- 4 interne Register für Fließkommazahlen – Intern bedeutet hierbei, dass diese nur für Zwischenrechnungen und die Übertragung von Daten in die nächste Runde des IPD benutzt werden und nicht von außerhalb des evolvierten Programms ausgelesen werden.
- 4 interne Register für ganze Zahlen (Typ „Integer“).
- 4 interne Register für boolesche Werte.
- 2 Eingaberegister für boolesche Werte – In das erste Register wird vor einem Zug gespeichert, ob es sich um den ersten Anfangszug handelt und falls dem nicht so ist, wird im zweiten Register zusätzlich der letzte Zug des Gegners gespeichert.
- 1 Ausgaberegister für boolesche Werte – Aus diesem wird nach dem Ablauf des evolvierten Programms ausgelesen, ob die repräsentierte Strategie defektieren soll oder nicht.

Die Benutzung von linearem genetischen Programmieren anstelle der üblicheren Variante mit baumartiger Repräsentation hat den Vorteil der Vereinfachung des Beschreibens von Schleifen und der einfachen Möglichkeit der Übernahme von Teilergebnissen in die nächste Runde durch Beibehaltung der Registerinhalte. Weiterhin haben die in der Praxis oft anzutreffenden Strategien jeweils nur eine geringe Speicherkapazität, dies wird durch das lineare Modell viel besser repräsentiert.

Es folgt eine Liste der verfügbaren Befehle:

- **Arithmetische & logische Operationen:** *Addition, Subtraktion, Multiplikation, Division, Und, Oder, Nicht*
- **Benutzung von Konstanten:** *Laden einer Konstante, Laden einer zufälligen Zahl*
- **Vergleichsfunktionen:** *Setze falls Null, Setze falls größer Null, Setze falls kleiner Null* – Diese Befehle setzen eine boolesche Variable auf *true*, wenn ein angegebenes Kriterium erfüllt ist.
- **Datenfluss:** *Wert kopieren* – Kopiert auch Werte zwischen verschiedenen Registertypen.
- **Kontrollfluss:** *Bedingter Vorwärtssprung, Bedingter Rückwärtssprung* – Dabei wird die Information, ob der Sprung durchgeführt werden soll, aus einem booleschen Register ausgelesen.

Alle Befehle arbeiten mit allen Registertypen, die für den jeweiligen Befehl Sinn ergeben. Insgesamt gibt es damit 34 verschiedene Befehls/Register-Kombinationen. Um die Ausführung von Endlosschleifen zu verhindern, ist die Anzahl der Ausführungen des bedingten Rücksprungbefehls auf 100 (pro Runde des IPD) beschränkt. Wird die Anzahl überschritten, wird der Befehl ab der nächsten Ausführung einfach ignoriert. Die maximale Anzahl der Befehle pro Strategie ist 500.

9.2.1.1. Phasen der Evolution

Es soll nun ein Überblick über die Phasen der Evolution gegeben werden. Dabei werden die für die jeweiligen Phasen getroffenen Design-Entscheidungen beleuchtet.

9.2.1.1.1. Initialisierung Bevor die Evolution starten kann, muss die Population mit Individuen gefüllt werden. In diesem Fall wurde eine zufällige Initialisierung verwendet. Es wird zuerst zufällig (gleichverteilt) eine Länge zwischen 4 und 15 gewählt und dann so viele zufällige Befehle in ein jeweils neues Individuum eingefügt. Da die Programmlänge während des Laufs des evolutionären Algorithmus noch steigen kann, beschränkt dies die Komplexität der später erzeugten Individuen nicht.

9.2.1.1.2. Rekombination Bei der Rekombination wird mit einer Wahrscheinlichkeit von 60% das Kind mit einem Verfahren namens „Zwei-Punkt-Crossover“ [BNKF98] aus zwei zufällig gezogenen Eltern erstellt, andernfalls wird ein zufällig gewählter Elternteil dupliziert. Alle generierten Kinder durchlaufen danach noch die Mutation.

9.2.1.1.3. Mutation Im Gegensatz zu Repräsentationen fester Länge muss der hier eingesetzte Mutationsoperator etwas komplizierter sein, da die Anzahl der Befehle variieren kann. Es gibt insgesamt drei verschiedene Typen der Mutation:

- Löschen eines Befehls
- Ändern eines Befehls
- Einfügen eines Befehls

In den letzten beiden Fällen wird dabei ein Befehl stets vollständig ausgetauscht und nicht nur ein Operand geändert.

9.2.1.1.4. Selektion Wie bereits erwähnt, wurde der LGP-Ansatz für zwei verschiedene evolutionäre Umgebungen eingesetzt: Die Evolution gegen eine feste Menge von Strategien und die Koevolution. In beiden Fällen kann die Fitness eines Individuums einer Population jedoch durch eine Zahl ausgedrückt werden. Daher kann in beiden Fällen derselbe Selektionsoperator angewendet werden.

Hierfür wurde ein recht einfacher Selektionsoperator entwickelt, welcher es erlaubte, die Wahrscheinlichkeit, dass schlechtere Individuen in die nächste Generation übernommen werden, zu bestimmen und somit auch implizit den Selektionsdruck anzupassen. Eine solche „weiche“ Selektion ist immer dann von Vorteil, wenn die Evolution länger laufen soll und somit die Wahrscheinlichkeit des Verweilens in einem lokalen Optimum erhöht ist. Bei diesem Selektionsoperator wird für jedes Individuum der Fitnesswert bestimmt und daraus der „effektive“ Fitnesswert berechnet:

$$F_{\text{effektiv}} = F_{\text{real}} \cdot e^{\alpha \cdot \varphi} \quad (9.1)$$

Dabei ist α der Parameter der Randomisierung der Selektion und φ eine normalverteilte Zufallsvariable. Nach der Berechnung der „effektiven“ Fitnesswerte werden die Individuen mit den höchsten dieser Werte ausgewählt. Dabei stehen sowohl die Individuen der Elterngeneration, als auch die frisch erzeugten Kinder zur Verfügung.

Wie zu sehen ist, kann damit die Art der Selektion durch Variieren des Wertes von α angepasst werden. Ein Wert von $\alpha = 0$ entspräche der elitistischen Selektion, während bei sehr großer Wahl von α die Selektion immer ähnlicher zur zufälligen Auswahl von Individuen wird und damit effektiv keine Evolution stattfindet.

9.2.1.2. Selbstadaptation

Die Evolution eines als Liste von Befehlen dargestellten Programms ist wesentlich aufwändiger als es bei den klassischen Repräsentationen der Fall ist. Daher ist eine höhere Berechnungszeit zu erwarten. Um diesen Effekt etwas zu reduzieren wurde Selbstadaptation eingeführt. Dabei wurde ein Schema verwendet, welches einem oft bei Evolutionsstrategien verwendeten gleicht [Bäc95]. Jedes Individuum hat dabei seinen eigenen Wahrscheinlichkeitsvektor, der jeweils die Wahrscheinlichkeiten für das Einfügen eines Befehls, das Löschen eines Befehls sowie das Austauschen eines Befehls enthält. Vor der Mutation eines Kindes werden die Wahrscheinlichkeiten, die es grundsätzlich vom ersten Elternteil übernommen hat, mit exponentiell normalverteilten Zufallszahlen multipliziert und danach die Mutation mit diesen Wahrscheinlichkeiten durchgeführt. Da bei einem Individuum mit einer günstigen Mutationsrate auch die Wahrscheinlichkeit, dass seine Nachkommen eine gute Fitness haben, steigt, ist die Wahrscheinlichkeit der Fortpflanzung bei guten Mutationsratenvektoren größer und dadurch werden diese ebenfalls mit evolviert.

Um einen Effekt, welcher von den Evolutionsstrategien als Degeneration der Schrittweiten bekannt ist [Rud99], zu verhindern, sind die Mutationsraten jedoch eingeschränkt, sodass diese weder zu hoch werden können, was dann sehr einer zufälligen Suche gleichen würde, noch zu niedrig, was dafür sorgen würde, dass der evolutionäre Algorithmus in einem lokalen Minimum hängen bliebe. Tabelle 9.1 stellt die Grenzen für die Mutationsraten dar.

Tabelle 9.1.: Übersicht über den Wahrscheinlichkeitsvektor für die Selbstadaptation. Zu jedem Individuum der Population gehört genau ein Vektor \vec{p} . Sollte aufgrund der Kürze eines Individuums der minimale Wert für den Austausch eines Befehles über dem Maximum liegen, wird der Wert auf das Maximum gesetzt.

| Evolutionärer Operator | Wahrscheinlichkeitsgrenzen | | Startwert bei Beginn der Evolution |
|-------------------------|--|----------------|------------------------------------|
| | Minimaler Wert | Maximaler Wert | |
| Löschen eines Befehls | $\frac{10\%}{\text{Länge}(\text{Individuum})}$ | 15% | 1% |
| Austausch eines Befehls | $\frac{50\%}{\text{Länge}(\text{Individuum})}$ | 10% | 5% |
| Löschen eines Befehls | $\frac{10\%}{\text{Länge}(\text{Individuum})}$ | 15% | 1.5% |

9.2.2. ES-Ansatz

9.2.2.1. Motivation

Die Psychologie ist eine Wissenschaft, die das menschliche Erleben und Verhalten zu beschreiben, zu erklären und vorauszusagen versucht. Ihr vorrangiges Ziel ist es also, Gesetzmäßigkeiten hinsichtlich der Faktoren, die das Erleben und Verhalten beeinflussen und den Folgen dieses Verhaltens aufzudecken. In der Literatur finden sich zahlreiche Theorien und Modelle zur Wahrnehmung, dem menschlichen Erleben sowie der Motivation und Handeln. Diese so genannten Funktionsbereiche lassen sich ferner aus verschiedenen Perspektiven betrachten [MG01], [MG02]. So versucht die allgemeine Psychologie allgemein gültige Gesetzmäßigkeiten in den Funktionsbereichen aufzudecken, ohne soziale oder differentielle Aspekte zu thematisieren. Diese sind Gegenstand der Forschung innerhalb der Sozial- bzw. Persönlichkeitspsychologie. Während die Sozialpsychologie sich mit dem Erleben und Verhalten unter Einfluss sozialer Faktoren befasst, stellt die Persönlichkeitspsychologie Rahmentheorien und Modelle der Struktur des Menschen und seiner Interaktion mit der Umwelt auf. Die hier vorgestellte Herangehensweise ist durch einige dieser Theorien inspiriert, wobei insbesondere Überle-

gungen der Motivationspsychologie und Konzepte der Persönlichkeitspsychologie als Ausgangspunkte dienten [Hec89], [Zim88], [Ase96].

Das angestrebte Ziel, eine erfolgreiche Strategie für das IPD zu finden, wird durch Nachahmung des menschlichen Verhaltens zu erreichen versucht. Das bedeutet, dass zum einen die evolvierten Strategien ähnlich wie Menschen auf verschiedenartige Situationen flexibel reagieren und diese sogar zu ihrem Nutzen zu beeinflussen versuchen sollten. So handelt erfahrungsgemäß eine Person in freundlichen Umgebungen kooperativer als in unfreundlichen. Andererseits sollen interindividuelle Unterschiede der Persönlichkeit in Betracht gezogen werden. So werden voraussichtlich zwei Personen auf dieselbe Situation unterschiedlich reagieren bzw. in ihr agieren. Während der Eine nachgiebig auf eine Provokation reagiert, wird ein Anderer sich herausgefordert fühlen und zu aggressivem Verhalten tendieren. Um die angeführten Gedanken auf das IPD zu übertragen, soll eine Strategie in der Lage sein, zwischen freundlichen und unfreundlichen Kontrahenten zu unterscheiden. Weiterhin soll sie auch die Schwächen der Gegner erkennen und nach Möglichkeit diese zum eigenen Vorteil ausnutzen können. Hinsichtlich des zweiten Punktes sollen sich Strategien in ihren Grundeigenschaften unterscheiden können, insbesondere in Bezug auf ihre Aggressivität, Nachgiebigkeit oder Erinnerungsvermögen.

9.2.2.2. Beschreibung des Genotyps

Für das beobachtbare Verhalten einer Strategie ist eine Vielzahl von Parametern verantwortlich. Sie sollen in diesem Abschnitt eingeführt und beschrieben werden. Wie die Parameter zusammenwirken und sich in den Entscheidungsalgorithmus zur Bestimmung des Verhaltens einer *IPD – Strategie* einfügen, wird schließlich im nächsten Abschnitt näher erläutert.

Die verwendeten Parameter zur Beschreibung einer Strategie lassen sich in zwei Teilmengen unterteilen. Während die erste Teilmenge die *situationsunabhängigen Eigenschaften* einer Strategie umfasst, wird mit der zweiten eine Aktivierungsfunktion $M(x)$ beschrieben. Diese Aktivierungsfunktion modelliert die Art und Weise, wie eine Strategie die vergangenen Interaktionen wahrnimmt. Sie kann als eine situationsabhängige Komponente einer Strategie aufgefasst werden. Die Tabellen 9.2 und 9.3 geben eine Übersicht über die verwendeten Parameter, ihre Definitionsbereiche und die gültigen Intervallgrenzen.

Tabelle 9.2.: Auflistung der grundlegenden Eigenschaften einer *IPD – Strategie*, mit Definitionsbereichen und den gültigen Intervallen

| Parameter | Symbol | Definitionsbereich | Intervall |
|---------------------------|-------------|--------------------|-------------------------------|
| FirstMove | <i>move</i> | \mathbb{B} | $\{false, true\}$ |
| Cooperation-threshold | <i>Th</i> | \mathbb{R} | $[0, 1]$ |
| QuitCooperation-threshold | <i>QcTh</i> | \mathbb{R} | [Sucker's payoff, Temptation] |
| MemorySize | <i>Mem</i> | \mathbb{N} | $[1, 100]$ |

Der Parameter *move* beschreibt das Verhalten einer Strategie zu Beginn eines Spiels. Wie der Tabelle 9.2 zu entnehmen ist, kann der Parameter lediglich zwei Zustände annehmen. Der Zustand *true* signalisiert ein kooperatives Verhalten der Strategie im ersten Zug, während bei der anderen Belegung die Strategie defektieren wird. Der nächste Parameter *Th* markiert einen Schwellwert für ein kooperatives Verhalten im Verlauf eines Spiels. Wird dieser Schwellwert überschritten, so kooperiert die Strategie, ansonsten wird defektiert. *Th* variiert von Strategie zu Strategie, ist jedoch für eine bestimmte Strategie immer fest. Der Parameter kann einen beliebigen reellen Wert zwischen eins und null annehmen. Eine ähnliche Aufgabe kommt dem Parameter *QcTh* zu. Dieser charakterisiert einen Schwellwert für ständige Defektion. Wird also zu einem Zeitpunkt der Begegnung der Schwellwert

Tabelle 9.3.: Parameter zur Modellierung der Aktivierungsfunktion, ihre Definitionsbereiche und die gültigen Intervalle

| Parameter | Symbol | Definitionsbereich | Intervall |
|---|------------------------------------|--------------------------------|----------------------------|
| Hauptspitze | <i>main</i> | $\mathbb{R} \times \mathbb{R}$ | $[0, 1] \times [0, 1]$ |
| <i>Gestaltparameter</i> _{main} (links) | <i>shape</i> _{main left} | \mathbb{R} | $[0, 100]$ |
| <i>Gestaltparameter</i> _{main} (rechts) | <i>shape</i> _{main right} | \mathbb{R} | $[0, 100]$ |
| Menge der <i>Nebenspitzen</i> | <i>peaks</i> _(i) | \mathbb{N} | $i \in \{1..4\}$ |
| <i>Nebenspitze</i> | <i>peak</i> | $\mathbb{R} \times \mathbb{R}$ | $[0, 1] \times [TH + 0.1]$ |
| Menge der <i>Gestaltparameter</i> _i (links) | <i>shape</i> _{i left} | \mathbb{R} | $[0, 100]$ |
| Menge der <i>Gestaltparameter</i> _i (rechts) | <i>shape</i> _{i right} | \mathbb{R} | $[0, 100]$ |

QcTh unterschritten, wird die Strategie für die restliche Dauer der Interaktionen defektieren. Sie wechselt also ihr Verhalten zu der klassischen Strategie *ALLD*. Der Parameter dient hauptsächlich zum Schutz der Strategie vor sehr unkooperativen Gegnern. Da *QcTh* mit der durchschnittlichen Auszahlung aus vergangenen Interaktionen verglichen wird, darf *QcTh* Werte zwischen der minimalen und der maximalen Auszahlung, also zwischen der Auszahlung *Sucker's payoff* und *Temptation* der Auszahlungsmatrix annehmen (siehe Kap. 2.3.1). Der letzte Parameter der situationsunabhängigen Eigenschaften ist *Mem*. Er beschreibt das Erinnerungsvermögen einer Strategie, also die Anzahl der vergangenen Züge während eines Spieles, an die sich eine Strategie erinnern kann. Gilt beispielsweise *Mem* = 10, so wird die Strategie maximal die letzten zehn Interaktionen in die Berechnung ihres nächsten Zuges einbeziehen. Weiter zurückliegende Interaktionen werden demnach „vergessen“.

Die Parameter aus Tabelle 9.3 charakterisieren die angesprochene Aktivierungsfunktion $M(x)$ einer Strategie. Ihre Aufgabe ist es, die vergangenen Interaktionen zu bewerten und eine Empfehlung abzugeben, ob die Strategie im aktuellen Zug kooperieren bzw. defektieren soll. Die Empfehlung äußert sich in Form eines Aktivierungsgrades *degree*, der abhängig von der durchschnittlich erhaltenen Auszahlung und dem Verlauf der Aktivierungsfunktion ist. Hierbei gilt die Grundregel: je höher *degree*, desto höher die Wahrscheinlichkeit für eine Kooperation. Ein exemplarischer Verlauf der Aktivierungsfunktion ist in Abb. 9.1 zu sehen.

Der Parameter *main* und die Menge *peaks*_(i) definieren lediglich die positiven Ausschläge (Stützpunkte) der Aktivierungsfunktion $M(x)$, also die Spitzen in der Abbildung 9.1. Die Anzahl der Nebenspitzen variiert von Strategie zu Strategie. Eine gegebene Strategie kann also entweder keine oder bis zu vier weitere Nebenspitzen enthalten. Da die Parameter lediglich die positiven Ausschläge beschreiben, muss der Gesamtverlauf aus diesen wenigen Parametern noch generiert werden. Die Konstruktion erfolgt hierbei nach einer fest vorgegebenen Methode. Im ersten Schritt wird die Menge der Stützstellen $\mathcal{P} = main \cup peaks_{(i)}$ nach ihren Abszissenwerten sortiert. Anschließend wird jeweils zwischen zwei benachbarten Punkten $P_i = \{x_i, y_i\}$ und $P_{i+1} = \{x_{i+1}, y_{i+1}\}$ ein weiterer Stützpunkt P_{bottom} hinzugefügt. Die Koordinaten des neuen Punktes errechnen sich nach folgender Formel:

$$P_{bottom} = \left(\frac{x_{i+1} - x_i}{2}, \frac{Th}{2} \right) \quad (9.2)$$

Der neue Stützpunkt wird also in der Mitte zwischen zwei Vorhandenen platziert, wobei sein Ordinatenwert die Hälfte des Parameters *Th* beträgt. In Abbildung 9.1 ist das Ergebnis der Prozedur verdeutlicht. Der neue Stützpunkt ist mit der Bezeichnung Senke markiert. Aus der Menge \mathcal{P} generiert die Prozedur also eine neue, um die Stützpunkte P_{bottom} erweiterte, Menge \mathcal{S} . Des Weiteren werden an den Grenzen der Aktivierungsfunktion, also an den Stellen $X = 0$ bzw. $X = 100$ zwei weitere Stützpunkte $S_{first} = \{0, 0\}$ und $S_{last} = \{100, 0\}$ explizit hinzugefügt. Für die sortierte

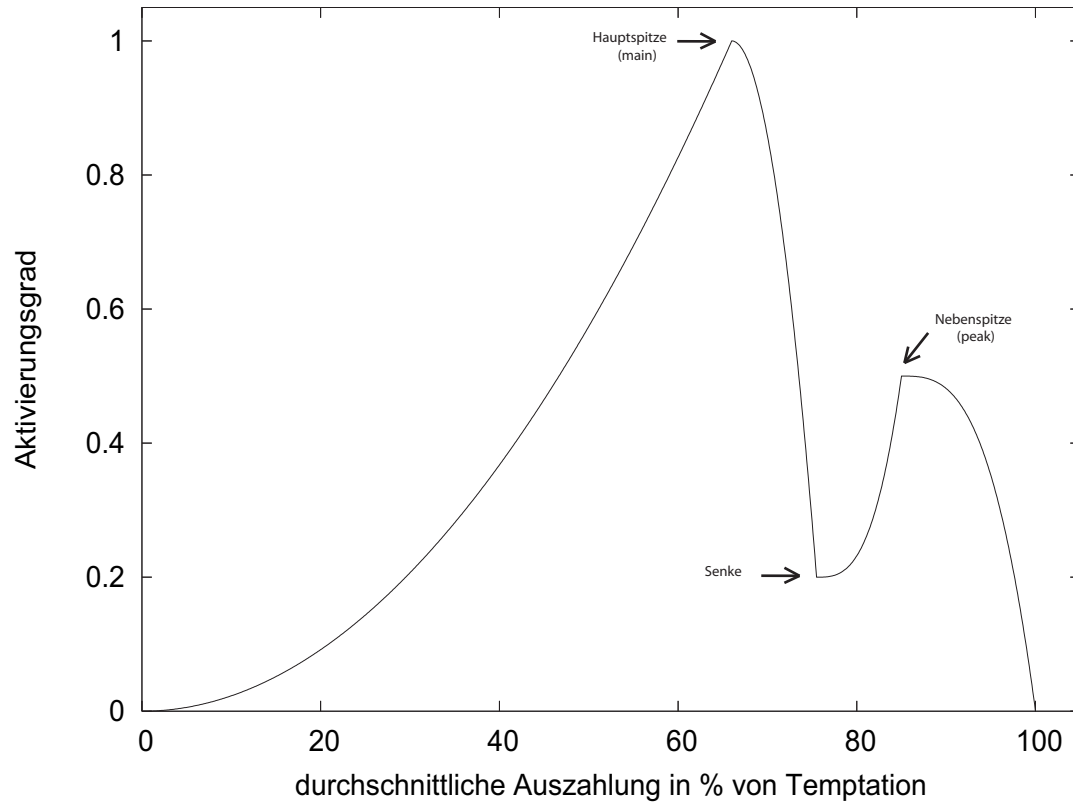


Abbildung 9.1.: Aktivierungsfunktion $M(x)$ als Bestandteil der situationsabhängigen Komponente einer Strategie

Menge \mathcal{S} ist also typisch, dass sich die Senken und Spitzen stets abwechseln. Für die Interpolation der Funktion zwischen zwei Stützstellen S_i und S_{i+1} werden die Gestaltparameter aus Tabelle 9.3 benötigt. Zu jeder Spitze bzw. Nebenspitze gehören jeweils zwei Gestaltparameter. Die eindeutige Zuordnung ist durch die Indizes sichergestellt. Diese Gestaltparameter bestimmen den Verlauf der Aktivierungskurve zwischen der entsprechenden Spitze und der links bzw. rechts benachbarten Senke. Die Interpolation der Funktion erfolgt hierbei nach folgender Vorschrift:

$$M(x) = (x - x_i)^{shape_i} * \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)^{shape_i} + y_i \quad (9.3)$$

Soll nun zu einem bestimmten Argument k der Funktionswert $y = M(k)$ berechnet werden, so erfolgt dies in mehreren Schritten: erstens wird das Intervall bestimmt, in welches das Argument k fällt. Es müssen also zunächst diejenigen Stützstellen $S_i = \{x_i, y_i\}$ und $S_{i+1} = \{x_{i+1}, y_{i+1}\}$ bestimmt werden, für die $x_i \leq k < x_{i+1}$ gilt. Anschließend muss der entsprechende Gestaltparameter $shape_i$ gesucht werden. Da einer der Stützpunkte S_i oder S_{i+1} eine Spitze bzw. Nebenspitze beschreibt, ist der Index des zugehörigen Gestaltparameters $shape$ bereits bekannt. Ungeklärt ist lediglich die Frage, ob der Parameter $shape_i$ aus der Menge der linksseitigen oder rechtsseitigen Gestaltparameter entnommen werden muss. Hier hilft folgende Überlegung: Gilt $y_i > y_{i+1}$ so handelt es sich bei dem Stützpunkt S_i um eine Spitze und bei S_{i+1} um eine Senke. Da nun $x_i \leq k < x_{i+1}$ gilt, muss also das rechtsseitige Intervall interpoliert werden. Folglich muss der Gestaltparameter $shape_i$ aus der rechtsseitigen Menge entnommen werden, also $shape_i$ *right*. Für den anderen Fall sind die Überlegungen analog.

9.2.2.3. Beschreibung des Phänotyps

Um zu verstehen, wie sich die beschriebenen Parameter auf das Verhalten der *IPD – Strategie* auswirken, wird an dieser Stelle der Pseudo-Code des Entscheidungsalgorithmus angegeben und anschließend erläutert.

```
(1)      Falls(erste Runde)
(2)          Spiele firstMove

(3)      sonst
(4)          Speichere die zuletzt erhaltene Auszahlung
(5)          Falls(zweite Runde)
(6)              Spiele Gegenteil von firstMove

(7)      sonst
(8)          Berechne die durchschnittliche Auszahlung average
(9)          Falls(average <= QcTH und Rundennummer>= Mem)
(10)             setze Flag SpiteMode = true

(11)     sonst
(12)         Berechne Aktivierungsgrad degree der Strategie
(13)         Falls(degree >= TH und spiteMode = false)
(14)             Spiele Kooperation
(15)         sonst
(16)             Spiele Defektion
```

Die Strategie eröffnet das Spiel mit dem durch *move* festgelegten Zug (Zeilen (1) und (2) im Pseudo-Code). Gilt hierbei *move = true*, so kooperiert die Strategie im ersten Zug, ansonsten wird

sie defektieren. In der folgenden Runde wird unabhängig von der erhaltenen Auszahlung das Gegenteil der ersten Runde gespielt (Zeilen (5) und (6)). Hat also die Strategie im ersten Zug kooperiert, so wird sie nun defektieren und umgekehrt. Darüber hinaus wird ab der zweiten Runde fortlaufend das Gedächtnis der Strategie mit den erhaltenen Auszahlungen gefüllt (5). Wird hierbei die Speicherkapazität, die durch Mem vorgegeben ist, überschritten, wird die älteste Auszahlung durch die zuletzt erhaltene ersetzt.

Nach der Eröffnungsphase, also ab der dritten Runde, kommt der eigentliche Bewertungsalgorithmus zum Einsatz. Die Strategie berechnet zunächst nach jeder Runde ihre durchschnittliche Auszahlung $average$ aus den vergangenen Interaktionen. Hierzu werden die relevanten Auszahlungen dem Gedächtnis entnommen und das arithmetische Mittel dieser Auszahlungen gebildet (8). Der Abgleich von $average$ mit $QcTh$ und der damit potentiell verbundene Wechsel zu einem $ALLD$ -ähnlichen Verhalten erfolgt erst im späteren Verlauf des Spiels ((9) und (10)). Der exakte Zeitpunkt ist von dem Parameter Mem einer Strategie abhängig. Im Allgemeinen muss jedoch das Gedächtnis einer Strategie vollständig mit Auszahlungen gefüllt sein, um einen Wechsel auf das $ALLD$ -ähnliche Verhalten bewerkstelligen zu können. Die Idee bei diesem Parameter war es, den evolvierten Strategien eine Schutzmaßnahme gegen unkooperative Kontrahenten zu geben. Jedoch sollte diese nicht in der frühen Phase des Spiels greifen und dadurch eine mögliche kooperative Begegnung frühzeitig gefährden. Wird also von dieser Option zu keinem Zeitpunkt Gebrauch gemacht, so wird durch den Wert $average$ im nächsten Schritt die wahrgenommene Situation durch die Strategie bewertet. Hierzu wird die aktuelle Punktausbeute $average$ auf das geschlossene Intervall $[0, 100]$ abgebildet. Dies entspricht einer Berechnung der Ausbeute in Prozent. Der neu erhaltene Wert wird als Argument in die Aktivierungsfunktion $M(x)$ eingesetzt (12). Der Aktivierungsgrad $degree$ der Strategie wird gemäß Gleichung (9.3) berechnet. Der resultierende Funktionswert $y = M(\frac{average}{Temptation})$ an der Stelle $average$ wird im nächsten Schritt mit Th verglichen (13). Überschreitet y den Schwellwert Th , so wird sich die Strategie kooperativ verhalten (14), andernfalls defektiert sie (16).

9.2.2.4. Beschreibung des evolutionären Verfahrens

Bei der Suche nach guten IPD -Strategien bzw. einer günstigen Konfiguration der Strategieparameter wurde auf die Methode der $(\mu + \lambda)$ -Evolutionstrategie mit Selbstadaptation der Schrittweiten zurückgegriffen [Sch94]. Die dem Optimierungsverfahren zugrunde liegende Idee ist es, den biologischen Prozess der Evolution, wie er in der Natur beobachtet werden kann, zu simulieren. Der Gesamttablauf des Optimierungsprozesses kann in folgende Schritte unterteilt werden:

1. Initialisierung
2. Rekombination
3. Mutation
4. Evaluation
5. Selektion

Die Schritte zwei bis vier werden im Laufe der Optimierung wiederholt durchgeführt, bis ein zuvor festgelegtes Abbruchkriterium erfüllt wird. Dieses ist in den meisten Fällen eine gewünschte bzw. erforderte Güte einer gefundenen Lösung. Alternativ kann das Verfahren nach einer bestimmten Anzahl von Iterationen oder einer gewissen Berechnungszeit abbrechen.

Initialisierung Zu Beginn der Optimierung wird die erste Elternpopulation generiert, insbesondere initialisiert der Algorithmus für jedes erzeugte Individuum die Objekt- und Strategievariablen. Hierbei werden die Objektvariablen zufällig mit Werten innerhalb der entsprechenden Intervallgrenzen

belegt. Im Gegensatz dazu nehmen die Strategievariablen eines Individuums alle denselben Anfangswert an.

Eine Modifikation des Originalalgorithmus besteht in der Organisation der Individuen einer Population. Die Individuen werden zu Teilgruppen zusammengefasst, deren gemeinsames Merkmal die Anzahl der Stützpunkte ihrer Aktivierungsfunktion ist. Die Entscheidung, eine Gruppierung der Individuen vorzunehmen, wurde im Hinblick auf die Rekombination getroffen. Durch diesen Schritt sollte ihr Implementierungsaufwand reduziert werden, da lediglich ähnlich aufgebaute Individuen miteinander rekombiniert werden.

Rekombination Die für den Algorithmus implementierte Rekombination ist an das *geometrical-crossover*-Verfahren von Michalewicz et al. angelehnt [MNM96]. Hierbei wird aus einer zufällig ausgewählten Menge von Elterindividuen jeweils ein Nachkommen erzeugt. Die Prozedur wird solange wiederholt, bis die zuvor festgelegte Anzahl von Nachkommen erreicht ist.

Mutation Alle durch die Rekombination erzeugten Nachkommen durchlaufen den für eine Evolutionsstrategie typischen Mutationsprozess. Zunächst werden alle Strategiekomponenten eines Individuums logarithmisch mutiert. Die Mutation erfolgt hierbei durch die Kombination der alten Schrittweite mit einem Wert, der für alle Strategievariablen eines Individuums gleich ist und einem anderen Wert, der für jede Strategievariable unabhängig gewählt wird. Anschließend erfolgt die Veränderung der Objektvariablen des Individuums. Hierbei wird zu dem aktuellen Wert der Objektvariable eine normalverteilte Zufallszahl hinzuaddiert.

Selektion Die Selektion einer neuen Generation erfolgt elitistisch, d. h. nur die besten Individuen einer Generation haben die Möglichkeit in die nächste übernommen zu werden. Da der Algorithmus die Plus-Variante einer Evolutionsstrategie implementiert, werden sowohl die Eltern als auch ihre Nachkommen für die Übernahme in die neue Generation berücksichtigt.

9.3. Umgebungen: anpassend vs. koevolutiv

9.3.1. Anpassende Evolution

Wie bei jedem Optimierungsverfahren muss die Art der Fitnessfunktion angemessen definiert werden, um aussagekräftige Ergebnisse erzielen zu können. Der Begriff der *anpassenden Evolution* steht im Zusammenhang mit dem IPD für das Optimieren der erzeugten Individuen an eine feste Menge von Gegenstrategien. Intuitiv bietet sich hierfür ein *Round-Robin*-Turnier als angemessene Fitnessfunktion an. In diesem spielt jedes Individuum ein IPD Spiel gegen alle Gegner und gegen sich selbst. Die Summe der im Turnier erzielten Punkte wird anschließend durch die Anzahl der gespielten Runden dividiert. Die so berechnete durchschnittlich erhaltene Auszahlung kann schließlich als Gütemaß des Verhaltens einer evolvierten Strategie während des Selektionsprozesses genutzt werden. Problematisch bei dieser Art der Fitnessberechnung ist die flexible Zusammenstellung der Menge der Gegenstrategien. Da sich jede mit jeder aufgenommenen bzw. weggelassenen Strategie die Fitnessfunktion leicht ändert, muss zunächst die Frage geklärt werden, welche Strategien in diese aufgenommen werden müssen. Die Auswahl der Strategien muss also mit dem Ziel, eine möglichst repräsentative Menge an Gegenstrategien zusammen zu stellen, erfolgen. Wie die Frage der Repräsentation gelöst wurde, ist Gegenstand des Kapitels 9.4.2.

9.3.2. Koevolution

In diesem Kontext bedeutet Koevolution, dass die Fitness der Strategien durch ein *Round-Robin*-Turnier bestimmt wird. Dies ist ein kritischer Unterschied zu den meisten anderen evolutionären

Algorithmen, da die Fitness einer Strategie nun von den anderen Strategien in der Population abhängt. Also kann sich die Fitness durchaus ändern, auch wenn der Genotyp einer Strategie über mehrere Generationen gleich bleibt, weil sich andere Strategien ändern.

Detailliert beschrieben passiert folgendes:

1. In einem Turnier spielt jede Strategie gegen jede andere.
2. Die Fitness wird als durchschnittliche Punktzahl über alle Spiele errechnet (akkumulierte Punkte geteilt durch die Anzahl der gespielten Runden).

Dieser Ansatz modelliert die natürliche biologische Evolution besser als evolutionäre Algorithmen, bei denen die Fitness eines bestimmten Individuums unabhängig von den anderen Individuen der Population ist, da die Individuen auch in der natürlichen Evolution nicht unabhängig voneinander sind. Im Gegenteil: Sie interagieren und beeinflussen sich gegenseitig. Zusätzlich zum Generieren von Turnierstrategien kann bei diesem Ansatz die Evolution der Kooperation beobachtet werden. Axelrod hatte Erfolg bei dem Versuch mit Hilfe von einfachen Repräsentationen kooperative Populationen zu generieren [Axe97]. Die Frage ist nun, ob dies auch bei komplexeren Repräsentationen funktioniert.

9.4. Versuchsvorbereitungen

9.4.1. Parameteroptimierung

Der Erfolg eines evolutionären Optimierungsverfahrens hängt von vielen Entscheidungen ab. Eine der Entscheidungen betrifft beispielsweise die korrekte Belegung der exogenen Parameter. Als exogene Parameter werden diejenigen Parameter bezeichnet, die durch den Anwender zu Beginn des Optimierungsprozesses festgelegt werden können. Im Falle der linearen genetischen Programmierung bzw. der Evolutionsstrategie wären dies unter anderem Parameter für die Größe der Elternpopulation, des Selektionsdrucks oder zur Steuerung der Schrittweiten während der Mutation. Da im Allgemeinen keine theoretisch fundierten Aussagen bezüglich der korrekten Parameterwahl für jeden Algorithmus und jede Problemstellung gegeben werden können, ist es angebracht und berechtigt günstige Parameterbelegungen experimentell noch vor Beginn des eigentlichen Optimierungsprozesses zu bestimmen. Für gewöhnlich läuft der Prozess der Parameteroptimierung folgendermaßen ab:

1. Spezifizierung der Intervallgrenzen für alle Parameter
2. Datenerfassung für alle festgelegten Konfigurationen
3. Auswertung der experimentell gewonnenen Daten

Diese drei Punkte werden im Rahmen der Parameteroptimierung mehrmals durchgeführt. Das Endergebnis besteht schließlich aus einer oder mehreren Konfigurationen von exogenen Parametern, die auf den verwendeten Optimierungsalgorithmus abgestimmt sind und somit die Wahrscheinlichkeit für qualitativ gute Resultate während der eigentlichen Optimierung erhöhen.

Für den zweiten und dritten der oben genannten Punkte existiert eine Vielzahl an denkbaren Verfahren. Eine verständliche Einführung in dieses Themengebiet bietet [BB03]. Hier, im Rahmen der in Kap. 9.2 vorgestellten Optimierungsverfahren, wurde auf die 2^k voll-faktorielle Methode zur Datenerfassung und das auf Regressionsbäumen basierende Verfahren zur Datenauswertung zurückgegriffen.

Bei dem 2^k voll-faktoriellen Versuchsaufbau zur Datenerfassung beschränkt sich die Belegungsmöglichkeit jedes der k exogenen Parameter lediglich auf zwei Werte. Folglich sind insgesamt 2^k unterschiedliche Konfigurationen möglich, die getestet werden müssen. Um eine statistisch abgesicherte Aussage bei der Datenauswertung über die Güte einer Konfiguration machen zu können, werden die einzelnen Konfigurationen mehrmals wiederholt.

Für die auf Regressionsbäumen basierende Datenauswertung werden die gewonnenen Daten zusammen mit den zugehörigen Konfigurationen zu einer Matrix aufbereitet. Ziel der Regression ist es, die exogenen Parameter nach ihrem Einfluss auf das Ergebnis des Optimierungsprozesses zu ordnen. Hierzu wird ein binärer Baum konstruiert, dessen Knoten die einzelnen Parameter repräsentieren. Je näher ein Parameter an der Wurzel des Baumes platziert ist, desto mehr Einfluss auf den Optimierungsprozess hat er. Die Auswertung der Daten wurde hier mit Hilfe des Statistikprogramms „R“ durchgeführt.

9.4.1.1. Ergebnisse der Parameteroptimierung für den LGP-Ansatz

Für den Ansatz des linearen genetischen Programmierens wurden insgesamt fünf exogene Parameter definiert und ihre Wirkung auf das Optimierungsergebnis in mehreren Durchläufen jeder Konfiguration bestimmt. Tabelle 9.4 fasst die Endergebnisse der Regressionsanalyse für die Parameteroptimierung zusammen.

Tabelle 9.4.: Die exogenen Parameter des Ansatzes des linearen genetischen Programmierens aus Kap.9.2.1. Die Spalte *Belegung* enthält die später bei der Ausführung des evolutionären Algorithmus genutzten Werte.

| Parameter | Symbol | Belegung | Bezeichnung in Abb. 9.2 |
|--|------------------------------|----------|-------------------------|
| Populationsgröße | μ | 1000 | A |
| Anzahl der Kinder pro Elter | ν | 3 | B |
| Selektionsdruck | <i>selDruck</i> | 0,3 | C |
| Länge der zweiten Periode | P_2 | 0 | D |
| Selektionsdruck in der zweiten Periode | <i>SelDruck_{P2}</i> | 2 | E |

Die Parameter μ und ν betreffen direkt die Menge der Individuen einer Generation. Während μ die Anzahl der Elternindividuen in einer Generation beschreibt, bestimmt ν das Reproduktionspotential eines Elternindividuum. In diesem konkreten Fall werden im Durchschnitt aus jedem Elternindividuum drei Nachkommen erzeugt. Der dritte Parameter *selDruck* ist für die Berechnung der effektiven Fitness eines Individuums relevant. Das Vorgehen wurde bereits näher in Kap. 9.2.1.1.4 erläutert. An dieser Stelle sei noch einmal wiederholt, dass dieser Parameter für die Randomisierung des Selektionsprozesses verantwortlich ist. Ein großer Wert von *selDruck* bedeutet hierbei starke Randomisierung. Im Gegensatz dazu verschärfen kleine Werte von *selDruck* die Auslese, was im Extremfall mit *selDruck* = 0 in einer elitistischen Selektion mündet.

Die letzten beiden Parameter P_2 und *SelDruck_{P2}* wurden im Hinblick auf eine Erweiterung bei der Fitnessevaluation ebenfalls in die Menge der exogenen Parameter aufgenommen. Diese wurde bei der Beschreibung des LGP nicht erwähnt, da die Parameteroptimierung ergab, dass die Erweiterung nicht sinnvoll ist. Sie betraf den Versuch, durch Einstreuung von Generationen, in welchen die Fitnessberechnung koevolutiv durchgeführt wird, die Rechenzeiteffizienz zu verbessern. Die Idee hierbei war, dass bei der Koevolution bei jedem Spiel für beide Gegner Fitnesswerte berechnet und verwendet werden können. Diese Idee brachte in der Praxis jedoch keinen Erfolg, was daran zu sehen ist, dass die optimale Parametereinstellung hier 0 ist.

Der zweite dazugehörige Parameter *SelDruck_{P2}* begrenzt die Anzahl der an einem Turnier teilnehmenden Individuen. Da bei der besten bekannten Parametereinstellung jedoch keine koevolutiven Generationen verwendet werden, ist der Wert dieses Parameters dort nicht wichtig.

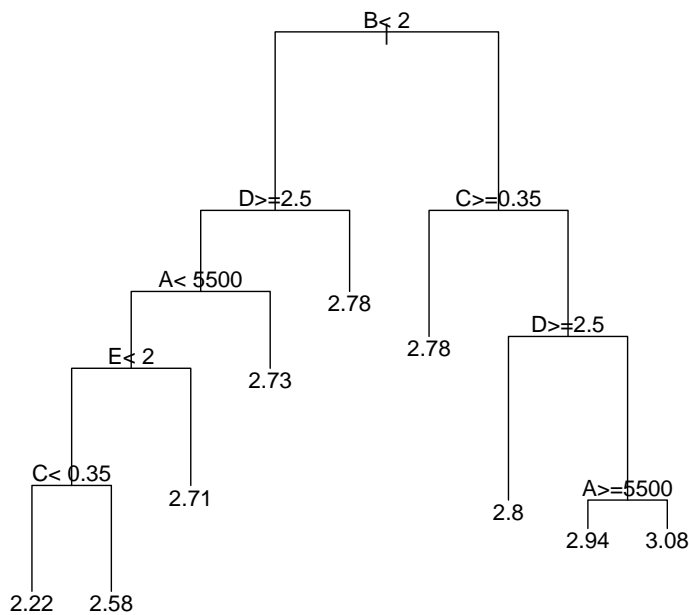


Abbildung 9.2.: Der Regressionsbaum für den Ansatz des linearen genetischen Programmierens. Jeder exogene Parameter wird einem Knoten des Baumes zugewiesen. Je größer der Einfluss des Parameters auf das Optimierungsergebnis ist, desto näher ist er an der Wurzel platziert. Der Abstieg von der Wurzel zu einem Blatt des Baumes kennzeichnet Parameterkonfigurationen. Dabei bedeutet ein Abstieg nach links, dass die im Knoten angegebene Bedingung erfüllt sein soll, bei einem Abstieg nach rechts hingegen nicht. An den Blättern stehen jeweils die bei einer Parametereinstellung zu erwartenden maximalen Fitnesswerte, welche den auf dem Weg von der Wurzel angetroffenen Bedingungen entsprechen.

9.4.1.2. Ergebnisse der Parameteroptimierung für die Evolutionsstrategie

Wie in Abschnitt 9.2.2 beschrieben, bedient sich dieser Ansatz einer Evolutionsstrategie als Optimierungsverfahren. Hierfür wurden insgesamt fünf exogene Parameter definiert und ihre Wirkung auf die Evolution und somit auf die Güte der gefundenen *IPD-Strategien* untersucht. Tabelle 9.5 bietet eine Übersicht über die Parameter und die für den eigentlichen Evolutionsprozess gewählten Belegungen.

Tabelle 9.5.: Die exogenen Parameter der Evolutionsstrategie des motivationspsychologischen Ansatzes aus Kap.9.2.2. Die Spalte *Belegung* enthält die später bei der Ausführung des evolutionären Algorithmus genutzten Werte.

| Parameter | Symbol | Belegung | Bezeichnung in Abb. 9.3 |
|--|-----------------|--------------------|-------------------------|
| Populationsgröße | μ | 10 | A |
| Anzahl der Kinder pro Generation | λ | 30 | B |
| Rekombinationspool | <i>marriage</i> | 2 | C |
| Anfangswert der Standardabweichungen zu Mutation der Strategievariable | $\sigma_{(0)}$ | 1 | D |
| Art der Rekombination | <i>Xrec</i> | <i>geometrisch</i> | E |

Die ersten zwei Parameter legen die Größe der Elternpopulation und die Anzahl der erzeugten Nachkommen fest. Ihr Quotient $\frac{\lambda}{\mu}$ wird auch als Selektionsdruck bezeichnet. Der nächste Parameter begrenzt die Anzahl der zufällig ausgewählten Elternindividuen, welche an einem Rekombinationsprozess und somit an der Generierung eines Nachkommens beteiligt werden. Bei der implementierten Evolutionsstrategie werden also aus insgesamt zehn Eltern jeweils lediglich zwei gezogen und miteinander rekombiniert. Wie die Rekombination vollzogen wird, ist von dem nächsten Parameter *Xrec* abhängig. Da zu Beginn der Implementierung völlig unklar war, ob eine diskrete Rekombination, also lediglich eine Art *Crossover*, ausreichend sei, wurde ebenfalls als Alternative die bereits oben erwähnte geometrische Rekombination nach Michalewicz [MNM96] implementiert. Wie an den Ergebnissen der Varianzanalyse abzulesen ist, scheint die Vermutung bezüglich der diskreten Rekombination korrekt gewesen zu sein. Der letzte exogene Parameter $\sigma_{(0)}$ betrifft die Initialwerte der Standardabweichungen. Wie Abb. 9.3 zu entnehmen ist, hat die Wahl des letzten Parameters $\sigma_{(0)}$ den größten Einfluss auf die Evolutionsstrategie. Gepaart mit der geometrischen Rekombination, welche im Regressionsbaum mit $E = 0$ gekennzeichnet ist, macht sie die Wahl der restlichen Parameter belanglos. Aus diesem Grunde wurden sie mit den in Tabelle 9.5 angezeigten Werten belegt, da diese Werte die unteren Grenzen bei der Parameteroptimierung bildeten und somit einen Vorteil an Rechenzeit mit sich bringen.

9.4.2. Strategiegewinnung und Entscheidungsbäume

Die wahrscheinlich wichtigste Fragestellung in Bezug auf die Evolution gegen eine feste Menge von Strategien ist, wie für in der Praxis vorkommende Gegner möglichst repräsentative Strategien für die Trainingsgegnermenge gefunden werden können. Durch diese Repräsentativität soll zum einen erreicht werden, dass die evolvierte Strategie in den typischen Turniersituationen günstig reagiert und insbesondere auch mit einem Gegner, welcher einem der Strategiemenge ähnlich ist, gut zusammenarbeitet oder, falls möglich, ihn sogar ausbeutet.

Axelrod hat in seinen Untersuchungen über die evolutionäre Gewinnung von Look-Up-Table-basierten Strategien [Axe97] die eingesandten Strategien des von ihm organisierten Turniers ver-

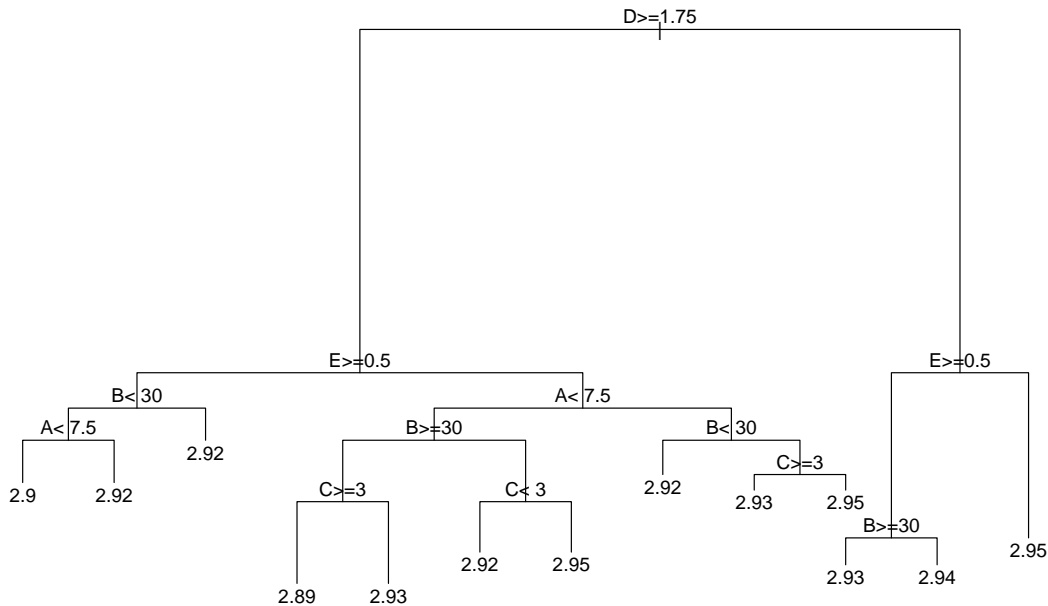


Abbildung 9.3.: Der generierte Regressionsbaum für die exogenen Parameter der Evolutionsstrategie. Jeder relevante Parameter wird einem Knoten des Baumes zugewiesen. Je größer der Einfluss des Parameters auf das Optimierungsergebnis, desto näher an der Wurzel ist er platziert. Der Abstieg von der Wurzel zu einem Blatt des Baumes kennzeichnet eine Parameterkonfiguration und ihr erwartetes Optimierungsergebnis.

wendet. Er stellte fest, dass sich das Gesamtergebnis von jeder Strategie in diesem Turnier mit nur geringer Abweichung aus dem Abschneiden der Strategie gegen acht bestimmte Gegner aus diesem Turnier erklären ließ. Daraus schloss er, dass diese auch gut als Trainingsgegner für den Evolutionsprozess geeignet seien.

Da für diese Untersuchung ebenfalls eingesandte Strategien des von der PG474 an der Universität Dortmund veranstalteten Turniers zur Verfügung standen, werden diese bei der anpassenden Evolution auch eingesetzt, jedoch vollständig (also nicht nur eine Auswahl). Da die meisten Strategien von Laien auf dem Gebiet des IPD verfasst wurden, sind diese aber als alleinige Referenzgegner sicherlich nicht ausreichend.

Für die Gewinnung zusätzlicher repräsentativer Strategien ist es also sinnvoll, die zuletzt durchgeführten größeren Turniere zu betrachten. Hier eingesandte Strategien können in identischer oder ähnlicher Form wieder eingereicht werden. Nun werden die Strategien selbst meist nicht veröffentlicht, jedoch enthalten die Turnierprotokolle die genauen Zugfolgen der Spiele aller Strategien. Es liegt deshalb nahe, mit Hilfe maschineller Lernverfahren Strategien zu erzeugen, welche sich so verhalten, dass diese Turnierprotokolle auch von ihnen stammen könnten und diese somit die eigentlichen Turnierstrategien imitieren. Dies wird natürlich im Allgemeinen nicht vollständig funktionieren, ist aber auch nicht zwingend nötig, da für die Benutzung als Trainingsgegner in der Evolution ja kein exaktes Verhalten reproduziert werden muss, sondern auch ein ähnliches Verhalten reicht.

Bei der Auswahl des maschinellen Lernverfahrens bieten sich insbesondere Entscheidungsbäume als Repräsentation an. Dabei verweisen die Knoten auf jeweils einen der letzten zehn eigenen Züge oder der Züge des Gegners. Um den nächsten Zug einer durch einen Entscheidungsbaum repräsentierten Strategie zu ermitteln, wird dieser von oben nach unten durchlaufen. Bei jedem Knoten wird der referenzierte Zug ermittelt und dementsprechend der mit D (wie Defektion) oder C (wie Kooperation) bezeichnete Ast verfolgt. Für den Fall, dass der referenzierte Zug nicht existiert, da sich das Spiel noch am Anfang befindet, gibt es einen Ast, welcher mit N markiert ist. Am Ende des letzten durchlaufenen

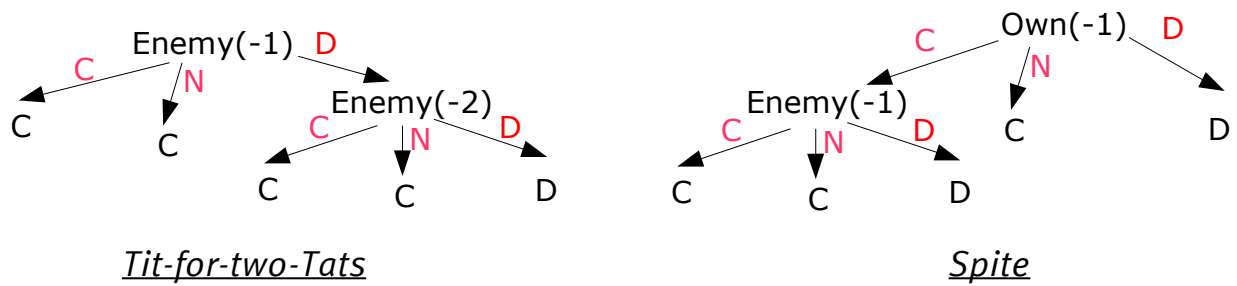


Abbildung 9.4.: Zwei Beispielentscheidungsbaume für zwei klassische Strategien - $Enemy(-2)$ bezeichnet dabei beispielsweise den vorletzten Zug des Gegners.

Tabelle 9.6.: Ausschnitt aus der Beispieltrainingsdatentabelle für die Strategie $PerNasty$ (zu imitierende Strategie), welche ein Spiel gegen TFT enthält. Zur besseren Übersicht sind nur die letzten drei statt zehn Züge dargestellt.

| Eigene letzte 3 Züge ($PerNasty$) | | | Gegnerische letzte 3 Züge (TFT) | | | Eigener nächster Zug |
|--|-----|-----|--|-----|-----|----------------------|
| -3 | -2 | -1 | -3 | -2 | -1 | |
| N | N | N | N | N | N | D |
| N | N | D | N | N | C | D |
| N | D | D | N | C | D | C |
| D | D | C | C | D | D | D |
| D | C | D | D | D | C | D |
| ... | ... | ... | ... | ... | ... | ... |

Astes befindet sich dann die getroffene Zugentscheidung, also entweder C oder D. Abbildung 9.4 stellt zwei solche Bäume dar.

Um aus dem Turnierprotokoll nun einen Entscheidungsbaum zu generieren, werden aus diesem zuerst Lerntabellen für alle Strategien erzeugt. Diese enthalten Datensätze, welche angeben, wie die Strategie bei den gegebenen letzten zehn eigenen und gegnerischen Zügen reagiert hat. Tabelle 9.6 stellt dar, wie ein aus einem Turnierprotokoll generierter Lerndatensatz aussehen kann. Danach wird ein $C4.5$ -Algorithmus [WE00] verwendet, um den Entscheidungsbaum zu generieren. Dabei werden 5% der Datensätze als Testmenge zurückgehalten.

Obwohl sich sehr viele Strategien durch solche Entscheidungsbaume darstellen lassen (z. B. TFT , $ALLC$, $Spite$), trifft dies nicht für alle zu (z. B. $Random$, $Joss$, $GenerousTFT$). Bei diesen Strategien wird der $C4.5$ -Algorithmus zwar einen Entscheidungsbaum generieren, dieser wird aber aufgrund der Randomisierung der eben genannten Strategien zwangsläufig Fehler machen, d. h. bei Eingabe der letzten zehn eigenen und gegnerischen Züge einer Zeile der Testmenge eine andere Antwort geben als der dort angegebene Zug. Um nun festzustellen, ob ein auf diese Weise aus einem Turnierprotokoll erzeugter Entscheidungsbaum die Strategie ausreichend gut imitiert, soll der Klassifikationsfehler der Testmenge betrachtet werden, d. h. es wird gemessen, welchen Anteil die Zeilen der Testmenge haben, bei denen der Entscheidungsbaum die falsche Antwort liefert. Es wird also ein maximal zuläs-

siger Klassifikationsfehler festgesetzt und alle Entscheidungsbäume, die bezogen auf ihre Testmengen einen Klassifikationsfehler größer als dieser maximal zulässige Fehler haben, werden nicht in die Referenzstrategiemenge hinzugefügt. Diese Fehlergrenze wird dabei so bestimmt, dass die bekannten klassischen Strategien, welche sich aufgrund von Randomisierung nicht durch Entscheidungsbäume abbilden lassen, gerade nicht mehr in die Referenzgegnermenge übernommen werden. Diese Vorgehensweise sorgt natürlich nicht dafür, dass genau die Strategien, welche sich durch Entscheidungsbäume darstellen lassen, auch tatsächlich in die Referenzgegnermenge übernommen werden. Diese Exaktheit ist aber auch gar nicht notwendig, da ja nicht genau die Strategien des Turniers dargestellt werden müssen, sondern für die Benutzung als Referenzgegner auch sich einfach nur ähnlich verhaltende Strategien ausreichen.

Im Übrigen kann dieses Teilproblem nicht exakt gelöst werden, da die im Turnier teilgenommenen Strategien noch zusätzliche Verhaltensweisen aufweisen können, die aufgrund der dort anzutreffenden Gegner aber nie gespielt wurden und somit im Turnierprotokoll auch gar nicht auftreten.

Für diese Versuche wurden die Turnierprotokolle der Turniere eins und vier der CIG2005 verwendet. Dabei traten insgesamt 176 verschiedene Strategien an. Zusätzlich zu den auf diese Weise erhaltenen Strategien bestand die Referenzstrategiemenge aus weiteren 18 klassischen in der Literatur vorkommenden Strategien, wie z. B. *GRADUAL*, *TfTT* und *PerNasty* sowie den 36 der 56 bereits erwähnten zu dem Turnier der Projektgruppe 474 der Universität Dortmund eingeschickten Strategien.

9.4.2.1. Ergebnisse

Der erste Schritt war die Bestimmung des maximal zulässigen Klassifikationsfehlers. Zusätzlich zu dem vorhandenen Satz an klassischen Strategien wurde aufgrund ihrer Einfachheit noch eine weitere Strategie aus dem PG474-Turnier berücksichtigt, welche eindeutig nicht durch Entscheidungsbäume darstellbar ist. Diese Strategie namens *Gralla50* spielt abwechselnd immer 50 Züge Kooperation und 50 Züge Defektion.

Ein diese Strategie imitierender Entscheidungsbaum könnte also eine Gestalt haben, in welcher die imitierende Strategie ihren letzten Zug immer wiederholt. Dies führt nur am Übergang zwischen Phasen der Defektion und der Kooperation zu Klassifikationsfehlern, so dass bereits dieser triviale Entscheidungsbaum nur eine Fehlerquote von 2% hat.

Interessanterweise ist *Gralla50* auch diejenige nicht durch Entscheidungsbäume vollständig darstellbare Strategie, welche beim Versuch, sie durch Entscheidungsbäume darzustellen, die niedrigste Fehlerquote von nur 1,3732% aufweist. Aus diesem Grund wird diese Prozentzahl des zulässigen Klassifikationsfehlers die Grenze darstellen.

Von den 176 Strategien, die an den Turnieren eins und/oder vier der CIG 2005 teilgenommen haben, konnte dabei in insgesamt 144 Fällen aus dem Turnierprotokoll ein Entscheidungsbaum generiert werden, dessen Klassifikationsfehler unter der eben festgelegten Grenze lag.

Diese 144 Strategien standen somit als Beispielgegner zusätzlich zu dem zum PG474-Turnier eingesandten Strategien und den klassischen Strategien zur Verfügung.

9.4.3. Trainings- und Testmenge

Tabelle 9.7 enthält eine Auflistung aller Strategien der verwendeten Trainingsmenge, Tabelle 9.8 eine Auflistung der Testmenge. Durch Verwendung dieser beiden Mengen kann später sowohl geprüft werden, wie gut die Evolution funktioniert hat (Platz in der Trainingsmenge) als auch vermutet werden, wie robust die erzeugten Strategien sind (Platz in der Testmenge). In der Testmenge wurden übrigens keine imitierten Strategien verwendet, diese befinden sich ausschließlich in der Trainingsmenge.

Tabelle 9.7.: Übersicht über die Trainingsmenge

| Typ der Strategien | Namen der Strategien |
|--------------------------------------|--|
| Klassische Strategien | <i>GRADUAL, TFT, HardTFT, SlowTFT, GoByMajority, TfTT, Pavlov Spite, Tester, Mistrust, ALLC, PerKind, HardMajority, Random, Prober, Joss, PerNasty, ALLD</i> |
| Strategien aus dem Turnier der PG474 | <i>Gralla50, Sebrassi IPD I, NilsTFT, Rasumichin, FJ Last_Twen Tit_for_tat with TfT-Recognition, StrategyByBas, Sauer1, Dirk Alpha, Meister, Sklave, Resistance is futile, ChristianTFT, Mugging Through, w-Random, Boris Konrad 2, Verzögerte Bestrafung, THModGradual1, Zweiundvierzig, MN2, Gartmann1, Dreiundzwanzig, TitForTat Plus1, TBattle Version 4, Der Zorn Gottes, SchwallisGradual, S-TFT, S-TFTT, Netzer, Delling, evaWobeGa, adamone, Deep, Minor Eisberg, Green Shark, Tasty Fish, Simon, Betonschuh, Die Macht von Grayskull</i> |
| Immitierte Strategien | <i>LOOKDB, LORD, LSF, Mem1, Mem2, Mod, NDCIUPA, NEG, NewForg1, NewForg3, NewForg4, OJ, Orac, PEON, PRObbary, PRObbaryL2, PRObbaryL3, PRObbaryL4, PRObbaryM2, PRObbaryM5, RANB, SLOS10, SLOS11, SLOS12, SLOS13, SLOS14, SLOS15, SLOS16, SLOS17, SLOS18, SLOS19, SLOS2, SLOS20, SLOS3, SLOS4, SLOS5, SLOS6, SLOS7, SLOS8, SLOS9, SOMETHING, STFT, ALLD, T4T, TCO3, TFT, TFTIa, TFTIb, TFTIc, TFTId, TFTIe, ALT, TFTIf, TFTIm, TFTIx, TFTIy, TFTIz, TFFT, TGTN, TTFT, TTFT 1, Yusuke Nojima, atft, highESTFT_classic, highESTFT_noise, lowESTFT_classic, lowESTFT_noise, ltft, mediumESTFT_classic, mediumESTFT_noise, tro1, Bingzhong Wang, C7D, CL*2, CL+2, CL-2, CLAS-, CLAS2, CLASS, CL/2, CL², CNGF, CNHM, DBSb, DBSd, DBSe, DBSf, DBSpl, DBStft, DBSx, DBSy, DBSz, DCIUPA, DETerminator, DETerminatorL2, DETerminatorL4, DETerminatorL5, DETerminatorL6, DETerminatorM2, DETerminatorM4, DETerminatorM6, DIIW, Forg1, Forg3, Forg4, Forg_a_1, GHCO_1, GHCO_2, GP_RDK, GRIM, GRIM ALLD TFFT ALLC RANDOM GRIM GRIM GBM, HCO, IMM01, IMS02, IMS03, IMS04, IMS05, IMS06, IMS07, IMS08, IMS09, IMS10, IMS11, IMS12, IMS13, IMS14, IMS15, IMS16, IMS17, IMS18, IMS19, IMS20, JJ01, Kaname Narukawa, KhuGRIMOne</i> |

Tabelle 9.8.: Übersicht über die Testmenge

| Typ der Strategien | Namen der Strategien |
|--------------------------------------|---|
| Klassische Strategien | <i>GRADUAL, TFT, SlowTFT, TfTT, Pavlov, ALLC, Random, ALLD</i> |
| Strategien aus dem Turnier der PG474 | <i>He-Man, Young Capone, Simon2, Major Eisberg, adamtwo, Schwal-li Faku, Gartmann2, MN1, THModGradual2, Boris Konrad 2, Dirk Beta, Sauer2, Raskolnikoff, FJ Last_Twen Tit_for_tat, Sebrassi IPD II, Throat, Netzer, Sebrassi IPD II, LateTFT, Späte Rache</i> |

9.5. Ergebnisse

9.5.1. Anpassende Evolution

9.5.1.1. LGP-Ansatz

Die Ergebnisse befinden sich in Tabelle 9.9.

9.5.1.2. ES-Ansatz

Bei der Suche nach guten Strategien für das IPD wurden für die festgelegte Konfiguration an exogenen Parametern mehrere Durchläufe durchgeführt. Die Abbildungen 9.5 - 9.7 zeigen die typischen Entwicklungen der Fitnesswerte während des Optimierungsprozesses. Die obere Kurve in jeder Abbildung charakterisiert den Verlauf der Fitnesswerte der besten Lösungen, also der erfolgreichsten Strategien, in einer bestimmten Generation. Die anderen beiden spiegeln die Entwicklung der durchschnittlichen Fitness der gesamten Population (mittlere Kurve) und die Fitnesswerte des schlechtesten Individuums in einer Generation (untere Kurve) wieder. Die darauf folgenden Abbildungen 9.8 - 9.10 auf Seite 126 zeigen die zugehörigen Standardabweichungen der Fitnesswerte einer Population in jeder Generation. Wie den Abbildungen zu entnehmen ist, liefert die verwendete Evolutionsstrategie stets ähnliche Resultate. Auffallend bei fast allen Durchläufen ist die Annäherung der Fitnesswerte an eine imaginäre Schranke von 2,63. Die besten der evolvierten Strategien erreichen diesen Fitnesswert rasch, während die restlichen Kurven diesem Trend mit einer gewissen Verzögerung folgen. Aus diesem Grund kann eine grobe Einteilung der Individuen in erfolgreiche und weniger erfolgreiche Strategien nicht vorgenommen werden. Dies sei nochmal anhand der Tabelle 9.10 auf Seite 132 verdeutlicht.

In der Tabelle ist für die zehn besten Strategien der Trainingsmenge – insgesamt umfasst sie 193 Strategien – die durchschnittlich erhaltene Auszahlung und die damit verbundene Platzierung eingetragen. Die Zahlen sind der Mittelwert aus fünf durchgeführten Turnieren. Es wird deutlich, dass der Optimierungsalgorithmus gute Resultate liefern kann. Die evolvierten Strategien belegen meist den ersten Rang. Ob es sich jedoch bei diesem Ergebnis um das globale Optimum handelt, kann nicht eindeutig geklärt werden. Um dies belegen zu können, muss für alle Gegnerstrategien die beste Verhaltensweise bekannt sein, also diejenige Verhaltensweise, mit der die meisten Punkte errungen werden könnten.

Eine Erklärung für das allgemein gute Abschneiden der evolvierten Strategien kann durch den Vergleich ihrer Fitnesswerte mit den Fitnesswerten der Trainingsmenge gefunden werden. Die evolvierten Strategien besitzen bereits in einem frühen Stadium der Evolution Fitnesswerte, die den besten der Trainingsmenge entsprechen. Aus diesem Grund ist ihre Dominanz verständlich.

Ein anderes Bild ergäbe sich jedoch, falls die Robustheit der evolvierten Strategien untersucht wird.

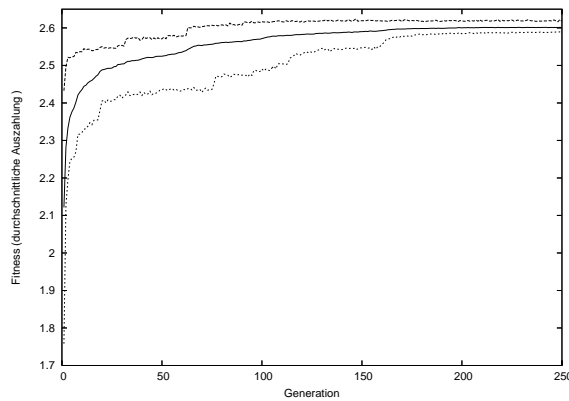


Abbildung 9.5.: Typische Entwicklung der Fitnesswerte während der „anpassenden Evolution“ mit dem ES-Ansatz. Die obere Kurve gibt den Verlauf der Fitness des besten Individuums in einer Generation wieder. Während die mittlere Kurve die durchschnittliche Fitness der gesamten Population widerspiegelt, markiert die untere Kurve die Entwicklung des schlechtesten Individuums pro Generation.

Das angestrebte Ziel war es, Strategien zu entwickeln bzw. mit einem evolutionären Suchverfahren zu finden, die in fast allen Umgebungen gut abschneiden. Die Tabelle 9.11 auf Seite 132 gibt die gemittelten Turnierergebnisse mit der festgelegten Testmenge wieder. Es ist deutlich zu erkennen, dass die Strategien deutlich abgeschlagen im unteren Drittel der Tabelle zu finden sind. Obwohl sich ihr Fitnesswert kaum verändert hat, reicht er nicht aus um die oberen Plätze der Tabelle zu erreichen. Aus diesem Grund kann für diesen Ansatz behauptet werden, dass die durch diesen Ansatz ermittelten Strategien sehr anfällig für Umgebungswechsel sind.

9.5.2. Koevolution

Zunächst wird die Evolution der Kooperation betrachtet. In Abbildung 9.11 ist für einige repräsentative Läufe die durchschnittliche Fitness gegen die Generation abgetragen.

Beim ES-Ansatz entwickelt sich recht schnell eine stabile Kooperation, d. h. eine durchschnittliche Auszahlung von 3 (entspricht gegenseitiger Kooperation). Beim LGP-Ansatz kann dies leider nicht beobachtet werden. Die durchschnittliche Auszahlung bleibt in der Nähe von 1 (d. h. gegenseitige Defektion). Nach Axelrod ist die Strategie *TFT* erst vorteilhaft, wenn eine gewisse kritische Grenze von kooperativen Individuen überschritten ist; daher wurde der Population ein Anteil von 5% *TFT* hinzugefügt. Wie in der Abbildung gezeigt wird ist dies ausreichend, um Kooperation entstehen zu lassen!¹

Vor dem Hintergrund dieser Ergebnisse wird nun das Abschneiden der evolvierten Strategien in einem Turnier gegen die Testmenge betrachtet: Hier zeigt sich klar die Überlegenheit der Strategien, die aus einer kooperierenden Population entnommen wurden (also aus einem Lauf der ES oder des LGP mit 5% *TFT*). Die aus der Population des LGPs mit 5% *TFT* entnommenen Strategien konnten, obwohl sie nicht auf bereits bekannte Strategien hin optimiert wurden, durchaus passable Plätze erzielen: Das beste untersuchte Individuum erreichte Platz sieben in der Testmenge. Viele Strategien lagen im Mittelfeld (Platz 14-17). Dagegen landeten die Strategien, die aus der unkooperativen Population entnommen wurden, meist nur auf dem letzten oder vorletzten Platz.

Strategien aus dem ES-Ansatz belegten meist die unteren Plätze (Platz 22-27 von 30). Eine Strategie erreichte allerdings auch den ersten Platz.

¹Genauer gesagt: In etwa der Hälfte aller Läufe des LGP mit Koevolution und 5% *TFT* entsteht gegenseitige Kooperation innerhalb der ersten 100 Generationen.

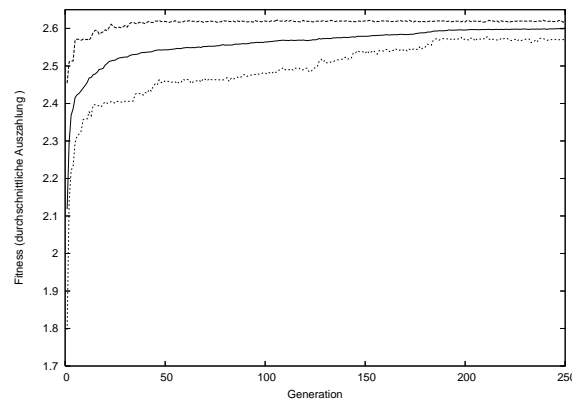


Abbildung 9.6.: Typische Entwicklung der Fitnesswerte während der „anpassenden Evolution“ mit dem ES-Ansatz (Die Abbildung zeigt einen zufällig ausgewählten Durchlauf). Die obere Kurve gibt den Verlauf der Fitness des besten Individuums in einer Generation wieder. Während die mittlere Kurve die durchschnittliche Fitness der gesamten Population widerspiegelt, markiert die untere Kurve die Entwicklung des schlechtesten Individuums pro Generation.

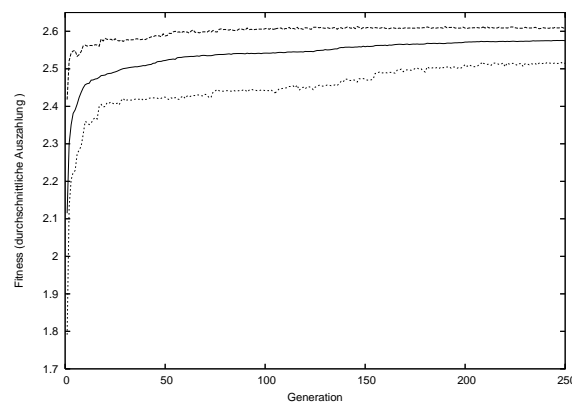


Abbildung 9.7.: Typische Entwicklung der Fitnesswerte während der „anpassenden Evolution“ mit dem ES-Ansatz (Die Abbildung zeigt einen weiteren zufällig ausgewählten Durchlauf). Die obere Kurve gibt den Verlauf der Fitness des besten Individuums in einer Generation wieder. Während die mittlere Kurve die durchschnittliche Fitness der gesamten Population widerspiegelt, markiert die untere Kurve die Entwicklung des schlechtesten Individuums pro Generation.

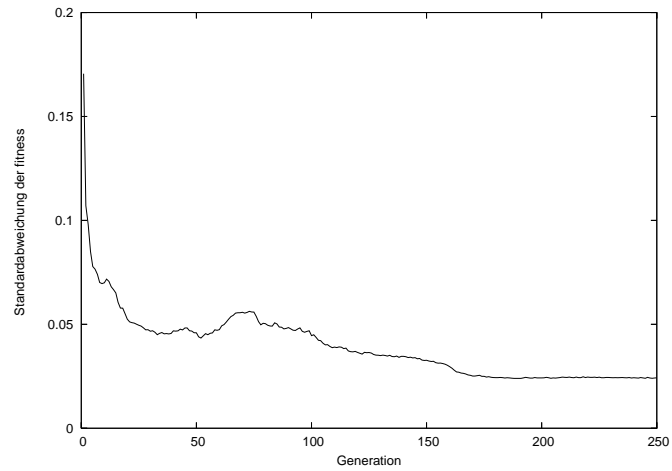


Abbildung 9.8.: Die Standardabweichung der Fitnesswerte in jeder Generation des Laufs aus Abb. 9.5. Es sei auf den rapiden Abfall der Standardabweichung und somit die schnelle Konvergenz des Algorithmus hingewiesen.

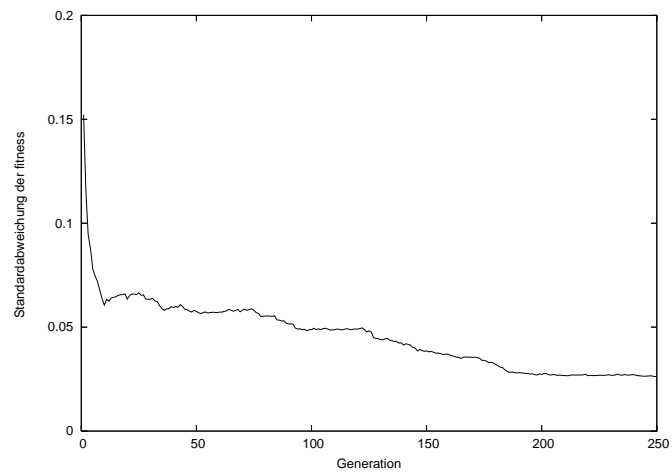


Abbildung 9.9.: Die Standardabweichung der Fitnesswerte in jeder Generation des Laufs aus Abb. 9.6. Es sei auf den rapiden Abfall der Standardabweichung und somit die schnelle Konvergenz des Algorithmus hingewiesen.

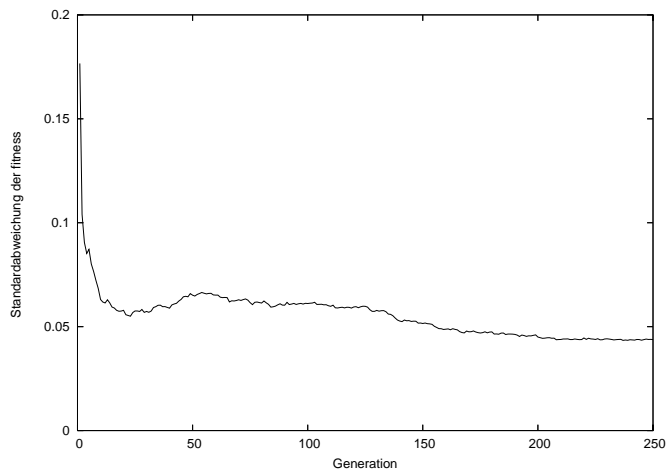


Abbildung 9.10.: Die Standardabweichung der Fitnesswerte in jeder Generation des Laufs aus Abb.9.7. Es sei auf den rapiden Abfall der Standardabweichung und somit auf die schnelle Konvergenz des Algorithmus hingewiesen.

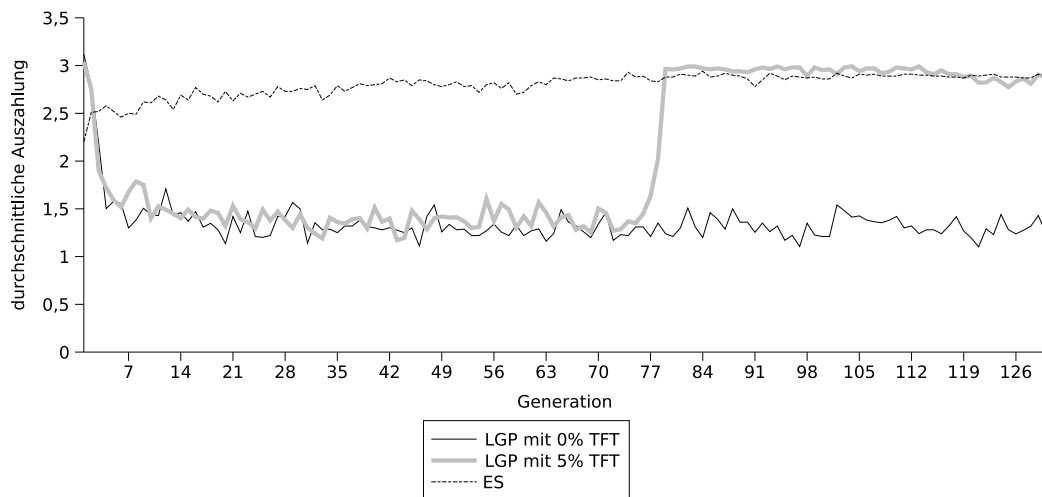


Abbildung 9.11.: Koevolution: Die Evolution der Kooperation

9.6. Analyse

9.6.1. Anpassende Evolution

9.6.1.1. LGP-Ansatz

Bei dieser Kombination sind interessante Ergebnisse zu sehen. Zwar erzielten einige Strategien sehr gute Ränge in der Testmenge, bei einem Großteil der erzeugten Strategien war dies jedoch nicht der Fall. Dies lässt mehrere verschiedene Schlussfolgerungen zu:

- Die Testmenge könnte trotz zufälliger Aufteilung der Gesamtstrategiemenge in Trainings- und Testmenge sehr unglücklich gewählt sein. Das Optimierungsziel wäre somit in der Trainingsmenge ein anderes als in der Testmenge und gute Ergebnisse demnach eher Glückssache. Diese Vermutung wird dadurch bekräftigt, dass die in der Testmenge guten Strategien wiederum in der Trainingsmenge nicht gut sind.
- Das Optimierungsproblem könnte in dieser Form auch zu schwer sein. Da der „Sprung“ von einer bereits guten zu einer noch besseren Strategie sehr groß ist, könnte die aufgewendete Rechenzeit um Größenordnungen zu klein sein, um mit genügender Wahrscheinlichkeit sehr gute Ergebnisse entstehen zu lassen.

Ein Erfolg des Ansatzes ist jedoch, dass die größere Menge darstellbarer Strategien, welche beim LGP gegenüber einfacheren Repräsentationen vorhanden ist, zumindest in einigen Fällen vom evolutionären Algorithmus auch tatsächlich genutzt wird. So kann z. B. die Strategie *Spite mit Vergabung im zweiten Zug* nicht durch Look-Up-Tables dargestellt werden.

Weiterhin ist die allgemein große Ähnlichkeit der evolvierten Strategien zur klassischen Strategie *Spite* zu bemerken. Da diese *Spite*-Varianten in der Trainingsmenge dabei recht gute Plätze erzielten, ist dies sicherlich darauf zurückzuführen, dass nachtragendes Verhalten tatsächlich eine gute Wahl beim Spiel gegen die Trainingsgegnermenge ist.

9.6.1.2. ES-Ansatz

Die erzielten Ergebnisse dieses Ansatzes bestätigen die Hypothese, dass es keine im Allgemeinen gute Strategie gibt. Dies wurde aus dem Vergleich der Resultate in der Trainings- und der Testmenge deutlich. Kritisch zu überprüfen wären folgende Fragen:

- Ist die Trainings- und Testmenge objektiv genug zusammengestellt worden?
- Ist die Repräsentation wirklich geeignet, um Strategien darzustellen?
- Sind die Operatoren des Optimierverfahrens optimal gewählt worden?

Der erste Punkt wurde bereits einen Abschnitt zuvor diskutiert und betrifft beide Ansätze zur Evolution von Strategien.

Bezüglich des zweiten Punktes muss festgestellt werden, dass der Ansatz trotz einer komplizierten Beschreibung eine kleine Menge der möglichen Strategien abdeckt. Im Gegensatz zum zuvor beschriebenen LGP-Ansatz, der die größtmögliche Freiheit bei der Modellierung einer Strategie zulässt, schränken die Vorbedingungen und der fest vorgegebene Berechnungsalgorithmus, ob zu kooperieren bzw. defektieren ist, den Suchraum sehr stark ein. Dies hat zwar zur Folge, dass recht wenige Strategien darstellbar sind, allerdings auch den Vorteil, den Suchraum schnell bewerten und somit ausreichend gute Lösungen schnell finden zu können. Die Ergebnisse bzw. der typische Verlauf des Optimierungsprozesses könnte als Hinweis auf diese Tatsache interpretiert werden.

Der letzten angeführten Frage können die Wahl bzw. die Einstellung der verwendeten Operatoren in der Evolutionstrategie als Antwort dienen. Insbesondere kann die elitistische Selektion die Vielfalt

der in einer Generation berechneten Lösungen zu schnell und zu stark einschränken. Deutlich wird dieses Verhalten an der gemessenen Standardabweichung der Fitnesswerte einer Generation (vgl. Abb. 9.8 bis Abb. 9.10). Als Gegenmaßnahme bietet sich hier eine verstärkte Nutzung der Randomisierung während des Selektionsschrittes an. Dieses Vorgehen würde zwar einen langsameren Anstieg der Fitnesswerte verursachen, andererseits vielen potentiell guten bzw. besseren Lösungen genügend Zeit geben sich zu entwickeln.

9.6.2. Koevolution

Hier lagen leider zum Zeitpunkt der Drucklegung dieses Endberichts noch keine endgültigen Ergebnisse vor.

Es stellt sich natürlich die Frage, warum beim LGP erst mit einer Starthilfe von 5% *TFT* Kooperation entsteht. In [CY05] wird spekuliert, dass dies eine grundlegende Eigenschaft indirekter Repräsentationen² ist.

Weiterhin legt der Erfolg der kooperativen Strategien die Gültigkeit der von Axelrod aufgestellten These nahe, dass erfolgreiche Strategien freundlich sein müssen.

9.7. Zusammenfassung

Im Rahmen der PG474 wurde die evolutionäre Gewinnung von Strategien an zwei grundsätzlich verschiedenen Repräsentationen mit den beiden bereits aus der Literatur bekannten Möglichkeiten für die Definition einer evolutionären Umgebung (anpassende Evolution und Koevolution) untersucht.

Die eine Repräsentation basiert auf genetischem Programmieren. Dabei können auch komplexere Strategien evolviert werden. Es hat sich jedoch gezeigt, dass trotz dieser Möglichkeit die erzeugten Strategien eher eine einfache Struktur hatten. Weiterhin erwies sich der Suchprozess hierbei als wesentlich schwieriger, was nicht nur zu einer stark erhöhten Laufzeit führt, wodurch die Qualität der Ergebnisse wegen der begrenzten Rechenzeit eingeschränkt ist, sondern auch in einer schlechten Evolution der Kooperation im koevolutiven Modell resultiert.

Die andere Repräsentation basiert auf einem psychologisch motivierten Ansatz. Hier konnte gezeigt werden, dass trotz der Einschränkung der darstellbaren Strategien sehr gute Ergebnisse evolviert werden können, die auch aufgrund der besseren Ausnutzung der Rechenzeit den mit dem LGP-Ansatz evolvierten Individuen in den meisten Fällen sogar überlegen waren. Auch die Evolution der Kooperation verläuft hier bemerkenswert gut.

Da sich beide Ansätze fundamental von den bisher vorgestellten Ansätzen auf diesem Gebiet unterscheiden, ergänzen sie diese in optimaler Weise. Zum Einen wurde in der Vergangenheit weniger auf Eigenschaften der menschlichen Entscheidungsträger, welche das IPD in praktischen Situationen spielen, eingegangen. Dies wurde durch das psychologisch motivierte Modell geändert. Zum Anderen wurde die Einschränkung der Darstellbarkeit von Strategien meist übersehen oder vollkommen anders problematisiert. So schrieben z. B. Beaufils, Delahaye und Mathieu [DMB98], dass umfassende Repräsentationen nie objektiv oder vollständig seien. Dies ist jedoch auf einen Ansatz, welcher auf genetischem Programmieren basiert, nicht mehr zutreffend.

Ebenfalls noch nicht in der Form vorgekommen ist die Technik der Strategieimitierung für die Gewinnung von Referenzgegnern für die Evolution. Aufgrund des in sicherlich jeder Projektgruppe auftretenden Zeitmangels besteht jedoch noch ein großes Potenzial für Verbesserungen.

Zu guter Letzt sei noch die konsequentere Form der Auswertung genannt. In vielen Publikationen über die evolutionäre Gewinnung von Strategien für das IPD wurde diese eher spärlich gehalten. So benutzten z. B. Darwin und Yao [DY95] die Spielergebnisse ihrer evolvierten Strategien gegen zufällige Strategien als Maß für die Robustheit dieser. Die Verwendung „echter“ Strategien, welche zum

²Indirekte Repräsentationen zeichnen sich durch eine komplexe Abbildung von dem Genotyp auf den Phänotyp aus.

im ersten Semester der PG veranstalteten Turnier eingesandt wurden, erlaubt jedoch eine viel realistischere Bewertung der Ergebnisse. Dies resultiert natürlich auch darin, dass die hier vorgestellten Ansätze auf den ersten Blick eine viel schlechtere Leistung erbracht zu haben scheinen.

Tabelle 9.9.: Ergebnisse der anpassenden Evolution beim linearen genetischen Programmieren. Um eine gute Vergleichbarkeit zu gewährleisten wurden jeweils zehn sehr gute Individuen und zehn Individuen mittlerer Fitness betrachtet.

| Anzahl Vorkommen | | Beschreibung der Strategie | Plätze in ... | |
|------------------------|---------------------|---|---------------|---------------------|
| mittlere Individuen | beste Individuen | | Testmenge | Trainings- menge |
| 0 | 5 | <i>Spite</i> | 14 – 16 | 15 – 21 |
| 1 | 0 | Zwei <i>Spite</i> -Strategien abwechselnd, es wird also für gerade und ungerade Züge unabhängig gemerkt, ob schon einmal defektiert wurde. | 1 | 10 |
| 0 | 1 | <i>Spite</i> mit Vergebung gegnerischer Defektion in der ersten Runde, danach Vergebung mit 50%-iger Wahrscheinlichkeit. Vergebung bedeutet, dass eine Defektion des Gegners folgenlos bleibt. Defektiert die Strategie jedoch einmal, so defektiert sie auch in den folgenden Zügen. | 15 | 12 |
| 1 | 0 | <i>Spite</i> mit $\frac{7}{8}$ Wahrscheinlichkeit für Vergebung. | 19 | 6 |
| 0 | 1 | <i>TFT</i> , wechselt jedoch zu <i>ALLD</i> nach zweimaliger Defektion des Gegners hintereinander. | 1 | 13 |
| 4 | 2 | <i>Spite</i> mit $\frac{1}{2}$ Wahrscheinlichkeit für Vergebung. | 9 – 16 | 11 – 13 |
| 4 | 0 | <i>Spite</i> , jedoch mit Vergebung im zweiten Zug (d. h. im ersten Zug darf der Gegner folgenlos defektieren). | 15 | 6 – 9 |
| 0 | 1 | <i>Spite</i> , jedoch mit einem „Freischuss“, d. h. einmal darf die Gegnerstrategie defektieren. | ??? | ??? |

Tabelle 9.10.: Gemittelttes Ergebnis der fünf Turniere mit Strategien aus der Trainingsmenge und einer nach dem ES-Ansatz evolvierten Strategie.

| Strategiename | Platzierung | Fitness | Standardabweichung der Fitness |
|---|-------------|---------|--------------------------------|
| <i>evolvierte Strategie (ES-Ansatz)</i> | 1 | 2.618 | 0.0325 |
| <i>Green Shark</i> | 2 | 2.523 | 0.3226 |
| <i>Netzer</i> | 3 | 2.449 | 0.0308 |
| <i>THModGradual1</i> | 4 | 2.434 | 0.0305 |
| <i>Muggling Through</i> | 5 | 2.422 | 0.0304 |
| <i>Boris Konrad 2</i> | 6 | 2.409 | 0.0304 |
| <i>DETerminatorL6</i> | 7 | 2.404 | 0.0302 |
| <i>CLAS2</i> | 8 | 2.404 | 0.0301 |
| <i>PRobbary</i> | 9 | 2.399 | 0.0301 |
| <i>PRobberyM2</i> | 10 | 2.398 | 0.0301 |

Tabelle 9.11.: Gemittelttes Ergebnis aus fünf Turnieren mit Strategien der Testmenge.

| Strategiename | Platzierung | Fitness | Standardabweichung der Fitness |
|---|-------------|---------|--------------------------------|
| <i>Späte Rache</i> | 1 | 2.851 | 0.040 |
| <i>Young Capone</i> | 2 | 2.841 | 0.040 |
| <i>Sebrassi IPD II (2)</i> | 3 | 2.836 | 0.041 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| <i>Throat</i> | 19 | 2.652 | 0.037 |
| <i>LateTFT</i> | 20 | 2.631 | 0.037 |
| <i>Pavlov</i> | 21 | 2.583 | 0.034 |
| <i>evolvierte Strategie (ES-Ansatz)</i> | 22 | 2.580 | 0.038 |
| <i>Major Eisberg</i> | 23 | 2.553 | 0.034 |
| <i>Sauer2</i> | 24 | 2.485 | 0.036 |
| <i>ALLC</i> | 25 | 2.276 | 0.032 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |

10. Fazit

Nachdem sich die Teilnehmer der Projektgruppe nahezu das ganze zweite Semester mit dem Erstellen der Konferenzarbeiten beziehungsweise der Generierung der Ergebnisse beschäftigt haben können für zukünftige Projektgruppen die folgenden Ergebnisse als Anregung dienen.

Grundsätzlich muss gesagt werden, dass es sich bei dem Erstellen von Konferenzarbeiten durch Kleingruppen einer Projektgruppe durchaus um eine semesterfüllende Aufgabe handelte. Bis zur Fertigstellung des Endberichts waren noch nicht alle vier Artikel fertig gestellt. Das Einsenden eines der vier Artikel wird auch erst zu einer späteren Konferenz erfolgen können. Als kurz nach Beginn des zweiten Semesters klar wurde, dass die Projektgruppe Konferenzarbeiten würde erstellen und einschicken können war den Teilnehmern der notwendige Arbeitsaufwand für ein solches Unternehmen nicht vollkommen bewusst. Zukünftige Projektgruppen sollten darauf ein besonderes Augenmerk legen und genügend Zeit nicht nur für die Experimente sondern auch und vor allem für das Schreiben an sich bereithalten. Die Sprachbarriere, die es bei internationalen Veröffentlichungen zu überwinden gilt, trug hier noch ein weiteres dazu. Auch der Aufwand des Layout-Prozess' war trotz der Verwendung von \LaTeX sehr hoch.

Neben dem hohen Arbeitsaufwand sind am Ende des zweiten Semesters jedoch nicht nur vier Konferenzarbeiten als Ergebnisse der Projektgruppe zu verbuchen. Durch die Nachforschung in internationalen und oft äußerst aktuellen Quellen wurde ein guter Einblick in das wissenschaftliche Arbeiten an einer Universität erarbeitet. In einigen Fällen kam es sogar zu Kontakten mit anderen Universitäten im In- und Ausland. Es wurden hier interessante Anregungen für eine weitere Beschäftigung mit dem Thema Gefangenendilemma aufgetan, die eventuell von einigen der Teilnehmer in ihrer späteren Laufbahn bearbeitet werden. Auch wenn es zum Zeitpunkt des Verfassens des Endberichts noch keine Rückmeldung zu den eingeschickten Konferenzarbeiten gibt, ist es vorstellbar, dass sie einen ersten Schritt in die Welt der Wissenschaft darstellen.

Abschließend ist zu sagen, dass die Aufgabe vom Umfang her einer Projektgruppe angemessen war. Das Schreiben von Konferenzarbeiten im Rahmen einer Projektgruppe kann weiterempfohlen werden.

Teil IV.

Zusammenfassung

11. Zusammenfassung

Das erste Semester wurde im Wesentlichen dazu genutzt, sich in die Thematik des IPDs einzuarbeiten und ein öffentliches IPD-Turnier auszurichten. Die dabei gemachten Erfahrungen (und auch die eingesandten Strategien) konnten im nächsten Semester weiter verwertet werden. Insbesondere wurde die Master-Slave Problematik erkannt und eine Turnierumgebung entwickelt, welche Grundlage für weitere Untersuchungen wurde. Am Anfang des zweiten Semesters wurde die Zielsetzung der PG in mehrere Fragestellungen aufgeteilt. Aus diesem Grunde haben sich die Teilnehmer der PG in drei Einzelgruppen unterteilt. Diese Einzelgruppen haben sich jeweils auf ein spezielles Thema festgelegt, diese möglichst ausführlich bearbeiteten und schließlich ein Paper zu ihrem Themengebiet geschrieben.

Erst in den Einzelgruppen wurde die konkrete Fragestellung (z. B. die Eliminierung des Vorteils der Master-Slave-Strategien gegenüber anderen Strategien) weiter strukturiert und ausgearbeitet. So trugen die Kleingruppen dazu bei, Ideen dynamischer und flexibler handhaben zu können.

Die Kleingruppe „Umgebungen“ hatte das grobe Ziel, alternative Turnierumgebungen und deren Einfluss auf die Platzierung der Strategien zu finden. Dabei war es bei genauerer Betrachtung wünschenswert, eine Umgebung zu konstruieren, welche Master-Slave-Strategien unterbindet bzw. deren Vorteil gegenüber anderen Strategien schwächt. Häufig sind diese Master-Slave-Strategien in einem IPD-Turnier nicht erwünscht, so dass sie ausgeschlossen werden sollten. Bei den anschließenden Bemühungen, potentielle Master-Slave zu entfernen, entstand die Umgebung *Jerusalem*. Diese lehnt an das Spiel „Reise nach Jerusalem“ an und vermeidet Master-Slave-Zusammenarbeit durch frühzeitigen „Rauswurf“ der Slave-Strategien. Es wurde noch nach einer Variante gesucht, in der Strategien von einem erwirtschafteten Punktepolster zehren können. Hieraus entstand die Umgebung *Jerusalem over all*. Beide Umgebungen waren nun in der Lage, unkooperative Slave-Strategien zuverlässig heraus zu filtern. Auch die von Axelrod vorgeschlagene betrachtete *ökologische Umgebung* war dazu geeignet.

Die Gruppe „Softwareagenten“ hatte das Ziel, agentenbasierte Anwendungen bzw. Simulationsmodelle für das IPD zu erarbeiten. Um aber das IPD auf die reale Welt abbilden zu können, musste erst ein geeignetes Simulationsmodell erstellt werden. Die Forderungen an so ein Modell waren Einfachheit und Durchschaubarkeit. Bei den Rechercharbeiten befand die Gruppe das Viscosity-Model von Axelrod für geeignet. Das Modell wurde mit der Möglichkeit zum „Punishment“ und mit der Möglichkeit zum Aufbau einer „Reputation“ erweitert, um eine bessere Abbildung auf die reale Welt zu garantieren. Die Teilgruppe „Punishment“ befasste sich in ihrer Arbeit mit der Möglichkeit zur Bestrafung von unkooperativem Verhalten. Die Teilgruppe „Reputation“ untersuchte die Auswirkung von Reputation, im Speziellen der Gruppenreputation. Als Ergebnis dieser Teilgruppen stellte sich heraus, dass beide Erweiterungen, Punishment und Reputation, das Kooperationslevel in dem gewählten agentenbasieren Simulationsmodell erhöht.

Die Gruppe EA hatte die Aufgabe, Strategien für das IPD durch evolutionäre Algorithmen zu entwickeln. Der Fokus lag hier insbesondere auf komplexeren Repräsentationen der möglichen Strategien. Diese komplexeren Repräsentationen wurden bisher noch sehr wenig untersucht. Es wurden zwei evolutionäre Algorithmen implementiert und untersucht: ein komplexer LGP-EA (eine Strategie ist hier Registermaschinenprogramm) und ein auf psychologischen Überlegungen beruhender ES-EA.

Die von den Kleingruppen erstellten Paper wurden am Ende des zweiten Semesters bei zwei Kongressen eingereicht, der CEC06 und der GECCO06.

Durch die verschiedenen Einzelgruppen wurde in dieser PG ein breites Spektrum möglicher Themen rund um das Gefangenendilemma abgedeckt. An den Ansätzen kann jedoch noch weiter gearbeitet

werden. Es wäre z. B. interessant, die vorgestellten neuen Umgebungen unter geänderten Randbedingungen zu testen, z. B. bei Erweiterungen wie „Rauschen“ oder „Multiplayer-IPD“. Ein weiteres interessantes Thema wäre die genauere Untersuchung der kooperativen Slaves und deren Verhalten in diesen Umgebungen, da dort nur zwischen diesen Slaves und Nicht-Master-Slave-Strategien unterschieden werden kann. Letztendlich könnte auch versucht werden, neue nicht-monotone Umgebungen zu finden, bei denen das erarbeitete Negativresultat nicht mehr greift und somit Master-Slave-Zusammenarbeit eventuell ganz verhindert werden könnte. Allerdings entspräche das nicht mehr den grundlegenden Konzepten einer Turnierumgebung. Die Teilgruppe „Softwareagenten“ hat, inspiriert durch die Teilgruppe „EA“ noch einen weiteren Ansatz einer möglichen agentenbasierten Simulationsumgebung entwickelt. Die Repräsentation der Strategien der einzelnen Agenten im Viscosity-Model (oder auch in Axelrods ökologischer Umgebung) könnte durch Entscheidungsbäumen umgesetzt werden. Es wäre interessant zu überprüfen, welche Strategien tatsächlich überleben und ob eventuell sogar eine Standardstrategie wie *TFT* entstehen kann. Des Weiteren könnte das Viscosity-Model in Kombination mit Reputation durch ein verblassendes Gedächtnis, momentan wird ein absolutes Gedächtnis verwendet, erweitert werden. Diese Veränderung könnte evtl. defektionsfördernd sein.

Der Hauptunterschied in der Gestaltung dieser PG zu der Organisation anderer PGs lag wohl in den großen Freiheiten, welche die PG-Teilnehmer bei ihrer Arbeit genossen. So bestand im ersten Semester kaum eine bestimmte konkrete Vorgabe, wie z. B. eine Strategie in einer bestimmten Umgebung zu programmieren und auszuprobieren. Es wurden der PG allerdings bestimmte Zielvorgaben gemacht, die jedoch variabel waren. So sollte das Originalexperiment von Axelrod wiederholt und zusätzlich mit bestimmten Änderungen noch einmal durchgeführt werden. Dies geschah im ersten Semester durch das Turnier am Campus-Fest bereits teilweise und im zweiten Semester wurden diese Vorgaben dann komplett erfüllt. Andere Vorgaben, wie z. B. das Rauschen zu untersuchen, wurden nicht oder nur zum Teil untersucht. Dafür haben sich die Teilnehmer, mit ausdrücklicher Unterstützung der Betreuer, eigene Aufgaben und Ziele gesetzt und diese auch umgesetzt. So war die Durchführung eines Turniers zwar ein Vorschlag der Betreuer, doch die Art der Ausführung, also ein öffentliches Turnier auf dem Campus-Fest, für das auch international Universitäten angeschrieben wurden, wurde nicht vorgeschrieben, sondern von der PG selbst beschlossen.

Auch im zweiten Semester kam der Grundgedanke, Paper zu erstellen und auf Konferenzen einzureichen, von den Betreuern. Jedoch die Anzahl der fertiggestellten (oder noch fertig zu stellenden) Paper hat selbst sie überrascht. Ebenso war es keine offizielle Vorgabe, die PG in mehrere Kleingruppen aufzuteilen, welche dann selbstständig an verschiedenen Themengebieten forschen sollten. Die im zweiten Semester untersuchten Gebiete waren nicht vorgegeben, sondern ergaben sich vornehmlich aus den Ergebnissen des ersten Semesters und den Wünschen und Vorstellungen der PG-Teilnehmer. Durch diese Aufteilung war es der PG möglich, ein wesentlich größeres Spektrum an Themengebieten abzudecken, als es einer einzelnen Gruppe möglich gewesen wäre. Ein weiterer Punkt, welcher sich sehr fördernd auf die Motivation auswirkte, war das bereits erwähnte interne Turnier. Auf jeder Sitzung wurde erneut bekannt gegeben, welche Strategie nun die erfolgreichste war. Dies spornte viele Teilnehmer zu weiteren Verbesserungen ihrer eigenen Strategie an.

Eine PG soll, wie im ersten Teil dieses Endberichts bereits geschildert, die so genannten *Softskills* der Teilnehmer verbessern. Da jeder Teilnehmer mehrmals im Verlauf der PG Sitzungsleiter war, konnte jeder die dafür nötigen Fertigkeiten entwickeln und weiter verbessern. Auch wurde durch das Protokollsystem oft verdeutlicht, wie wichtig es z. B. ist, einzelne Verantwortliche für jede Aufgabe zuzuteilen, da viele Arbeiten sonst wesentlich länger gebraucht hätten. Ein weiterer wichtiger Punkt, den die PG als sehr positiv hervorheben möchte, ist die Art der Betreuung. Es wurde von Betreuerseite weder zu sehr interveniert, noch wurde der PG völliger Freiraum gelassen. Diese Form der Betreuung wurde als sehr angenehm und hilfreich empfunden.

Die Art der PG-Gestaltung hatte jedoch selbstverständlich auch Nachteile. Vor allem wurde ein hohes Maß an Kommunikationsfähigkeit von jedem Teilnehmer gefordert, da sich die einzelnen Kleingruppen auch gegenseitig über die jeweiligen Fortschritte auf dem Laufenden halten mussten. Es

genügte nun nicht mehr, nur den Betreuern die Fortschritte aufzuzeigen, es musste auch den anderen Gruppen erklärt werden, wieso gerade dieser Ansatz gewählt wurde.

Allerdings ist es die Meinung der PG, dass vor allem die Kleingruppen ein hohes Maß an Motivation innerhalb der Gruppe schufen. Auch die Möglichkeit, selber an einem Paper zu schreiben und diese auf Konferenzen einsenden zu können, wird von allen Teilnehmern der Gruppe sehr positiv gesehen. Als Fazit lässt sich sagen, dass diese Art der PG-Gestaltung vielleicht unüblich ist, jedoch sind nach Meinung dieser PG die Vorteile dieser Art der Gestaltung wesentlich größer als die Nachteile.

Teil V.
Anhang

A. Environments for the IPD dealing with the Master-Slave Problem

New Environments for the Iterated Prisoner's Dilemma dealing with the Master-Slave Problem

Bastian Baranski, Thomas Bartz-Beielstein, Rüdiger Ehlers, Thusinthan Kajendran,
Björn Kosslers, Jörn Mehnen, Tomasz Polaszek, Ralf Reimholz, Jens Schmidt,
Karlheinz Schmitt, Danny Seis, Rafael Slodzinski,
Simon Steeg, Nils Wiemann, Marc Zimmermann

Abstract—Many publications have been published in the last years concerning the well-known iterated prisoner's dilemma (IPD). Though there are several papers dealing with changes of the payment matrix researches about tournament environments remained rare. Furthermore, if collaboration among strategies is not allowed, there still is a dilemma itself in creating fair tournaments due to the master-slave problem. This paper addresses the problem of creating fair and robust tournament environments for the iterated prisoner's dilemma, if the participation of master-slave strategies is not permitted. A discussion about the problem of detecting masters and their corresponding slaves is presented. In addition, we propose four new IPD environments, which are called *Jerusalem*, *Jerusalem over all*, PG474 and a variant of an ecological environment proposed by Axelrod. A detailed analysis and comparison of all environments lead to the conclusion that especially *Jerusalem* is able to lower the benefit of master-slave collaboration in a significant manner. Therefore *Jerusalem* seems to be well suited for tournaments where master-slave strategies are not allowed.

I. INTRODUCTION

The iterated prisoner's dilemma (IPD) is a well-known two-person game, which is particularly used for investigation in the evolution of cooperation and to illustrate individual conflicts. The goal is to find a good strategy for every given situation making the right decision. In 1984, Axelrod accomplished two *round-robin* tournaments, in order to find good strategies or maybe the best strategy for the IPD. Winner of the tournament was the simplest strategy, called *TIT FOR TAT*. This strategy cooperates in the first move from where on it plays the move the opponent played in his last turn. This surprising result arranged Axelrod to repeat the tournament. Again *TIT FOR TAT* won. Recently two tournaments in the context of the Congress on Evolutionary Computation (CEC) in 2004 and at the Computational Intelligence and Games (CIG) 2005 - fundamentally repetitions of the original experiment of Axelrod, but partly with several extensions like noise - led to problems with the classical *round-robin* environment. It appeared that participants submitted several strategies, which worked together. Among this submissions, a lot of strategies were only used to boost one so-called *master* strategy, which usually made his way to the top. So, with an effort of self-sacrificing, these strategies (*slaves*) played very bad against their master and defect against other strategies.

All authors are with the Department of Computer Science, Chair of Algorithm Engineering and Systems Analysis, University Dortmund, 44227 Dortmund, Germany (email: jens.schmidt@udo.edu).

This collaboration of strategies in a master-slave relationship may of course, if explicitly not desired, violate the entire spirit of an IPD tournament. And as long as their is nothing in the competition rules to preclude such master-slave strategies, the results of each IPD tournaments remain vulnerable [PM06].

If we follow this criticisms, the question arises, how can we prevent that master-slave strategies falsify the results of the IPD? There are several possibilities for this, starting from the bare exclusion of those strategies over the change of the payoff-matrix to the whole development of a new environment. Although there are many articles dealing with variations of the payoff-matrix [NAC92], [FOG93], [MUE88], researches focusing on the development of new tournament environment remained very rare [HC88], [DM95].

Topic of this article is the development and analysis of new tournament environments, which are robust against master-slave strategies. Robustness means that the advantages of the masters compared to normal strategies has to be reduced - if not eliminated. This contains in particular that the ranks of the remaining strategies remain unaffected.

The remainder of this article is arranged as follows: In section II a detailed discussion about the problem of detecting masters and slaves is presented. Subsequently, in section III four new environments are proposed. The experimental setup is presented in section IV and the experimental results of our comparisons are shown in section V. An analysis and discussion of the obtained results is depicted in the ensuing section VI. Finally, section VII summarize the results and give insights for further research.

II. PROBLEM

In this section a detailed overview of the working principle of master-slave and the resulting advantages and problems is presented. The result is a high level background that is necessary to develop robust and fair tournament environments.

A. The Working Principle of using Masters and Slaves

For demonstration purposes consider a set of single strategies with one master and one slave in addition. These strategies participate in a *round-robin* tournament. It is always necessary for collaboration among the master-slave strategies that both strategies play an identification sequence. This sequence is a predefined pattern consisting of cooperation (*C*) and defection

(D) moves. Both the length and the pattern itself are adjustable user-defined variables. If both strategies have recognized each other, the slave will continuously play cooperative while the master will be playing defect. Thereby the master receive an approximately optimal payment from the payoff-matrix, against what the slave gets nothing. In the case of not recognizing both strategies play a second, alternative strategy. In many cases it can be observed that the master plays *TIT FOR TAT* as the second strategy. Consequently, the slaves plays *uncooperative* (defect) against all other strategies.

All master-slave strategies are supported beyond that by the environments used so far.

B. Monotonic Tournament Environments

Usually every tournament consists of several rounds, in which the achieved points are added additively to the total amount of the strategies. For example Axelrods standard *round-robin* environment is additive after each game. There will be a problem resulting from this cumulative behavior so we need to define this behavior clearly:

For this we assume a slightly weaker proposition than addition: A tournament environment consists of several tournament rounds. We define a tournament environment *monotonic*, if and only if following conditions are satisfied:

- 1) All points scored in all tournament rounds that are incorporated in the total amount of points, are greater than 0 and are evaluated with respect to the standard payoff-matrix listed in [Axe84].
- 2) These calculated points are weighted with a strictly *monotonic* increasing function f and are incorporated in overall points in such a way that this amount grows strictly *monotonic*, too.

The simplest possible function f is the identical function. In such cases the tournament environment simply adds all points as in the case of the standard *round-robin* environment. Nevertheless the definition allows much more, for example it is possible to weight points with a strictly *monotonic* function f and then multiply the points or exponentiate them. Furthermore the function f can be altered through the tournament, e.g. dependent on the current tournament round in a way that less/more points are given at a later round. But it is important to note that the function f must not prefer a strategy, i.e. it can be dependent on the points strategies achieved but not on their name.

Since *monotonic* environments implement a common concept often made in tournaments, it is not a surprise to find all IPD environments with standard payoff-matrix, which we are currently aware of, in this class. In particular all environments, where positive points are summed up over all tournament rounds, as well as all of the proposed new environments in section III belongs to this class. We argue that each possible *monotonic* environment does not admit a solution to the problem of eliminating master-slave collaboration.

C. A negative result for all monotonic tournament environments

In every *monotonic* environment the overall points of all strategies are strictly *monotonic* increasing. For demonstrating purposes and without loss of generality we assume a strategy set consisting of several non-master-slave strategies except one single master and his slave. In addition there is a strategy in the set denoted as non-master strategy, which is absolutely identical to the second strategy of the master except that he has no slave. That is the same strategy without playing the identification pattern. We first will assume that all slaves are *uncooperative slaves*.

When tournament evaluation starts, the master takes a non-zero bonus from his slave, while the identical non-master strategy gets no bonus. Both strategies play exactly identical against all strategies but the master. That is why the master has an advantage to the non-master strategy from beginning. Since the tournament environment is *monotonic*, this advantage is never taken from the master. He will maintain this bonus until the end of evaluation. Therefore it is not possible to eliminate master-slave collaboration in this scenario. But we have not made any assumptions over the exact strategy the master is playing. So this argument effects all possible strategysets, saying that for every strategy there exists an identical strategy with at least one slave that performs better. So for the winner of every possible *monotonic* tournament there exists a master strategy, which outperforms this winner. Thus, there can't be a possibility to find a solution to the problem of eliminating master-slave collaboration. The other way round, someone can only eliminate master-slave cooperation, if he uses a *non-monotonic* environment. But this seems to contradict common concepts of tournaments speaking in general terms, for example it would be possible to take back points from a strategy that were achieved before. So if we remain in the class of *monotonic* environments we have to reformulate our goal:

The goal is to find a tournament environment, which minimizes master-slave collaboration as far as possible, i.e. minimizing the bonus the master gets from his slaves as far as possible.

Finally a few short remarks on the possibility to interrupt interactions between masters and slaves.

D. Identification patterns

Identification patterns have to be used by either master or slave to recognize each other. Usually this pattern is played at the beginning of a game, because then collaboration effects the most rounds of that game.

There is a point in considering these identification patterns as an option to eliminate master-slave collaboration, what is exactly what we want. Every master or slave has to use them for recognizing purposes, so all what remains to do is finding these and exclude the slaves from tournament. But unfortunately the possibilities of computing the identification patterns are very wide. First the length of each pattern is not limited. Furthermore the pattern has not be either connected nor deterministic nor played at the beginning of a game. So

the bits of this pattern could be distributed among arbitrary positions. Moreover it is possible to play several different patterns with a certain probability. In short, the number of different ways of collaborating with another strategy grows exponentially with the number of rounds played in the IPD. So, it is impossible in practice to check for master-slave interaction beforehand or during the game. We have to find another way to minimize the influence of master-slave strategies.

III. NEW ENVIRONMENTS

In the last section we have shown that the task of a fair tournament environment has to be the minimization of the bonus of the master. In this section we propose four new environments called *PG474*¹, *Jerusalem*, *Jerusalem over all* and in addition a slightly variant of the *ecological environment* originally proposed by Axelrod [Axe84]. The basic of each environment described below represents the *PG474 round-robin* tournament.

A. The *PG474 environment*

This environment represents a slightly modification of the *round-robin* tournament [Axe84]. In the *round-robin tournament* each strategy must play against all other strategies as well as against itself. Since strategies perform different when changing the length of game, i.e. the number of rounds (NR), we have to evaluate more than one run and compute the average of them. Reasonable should also be a minimum for all possible NR , because if NR is very small, all random moves of a possible stochastic strategy affects the whole tournament in a significant manner. Furthermore the exact position of the last round of the game should not be known, otherwise strategies will try to get bonus-points by defecting on the last round or the last but one. In order to do this, the actual NR is computed with the following geometrically distribution ($x \in \mathbb{N}$):

$$P(NR = gl_{min} + x) = gl_{min} + \frac{10}{gl_{min}} \left(1 - \frac{10}{gl_{min}}\right)^{x-1} \quad (1)$$

where gl_{min} denotes to the minimum length of each game and x denotes the excessive length over gl_{min} . This number is calculated once before the whole competition starts.

The tournament evaluation is calculated as the sum of all points, gained in each pairwise comparison. Here, and in all other environments, the payoff-matrix from [Axe84] is used. In case of two or more strategies resulting on the same place a deciding game will be performed, which consists of the involved strategies and a special strategy named *RANDOM*, which plays each move with a 50% chance cooperating.

In order to avoid effects of multiple slaves for one master, only two strategies for each participant is allowed. We will show that this restriction reduces the advantages of collaboration noticeable.

¹The *PG474* is a one year project group of 12 students of the University of Dortmund, Germany.

B. *Ecological Tournament*

The ecological tournament is an environment originally introduced by Axelrod. It consists of several standard *round-robin* rounds and intends that well-performing strategies in a tournament should take hold in the next round as well. It is called ecological because Axelrod pointed out that it is essentially the same mechanism found in nature where young individuals adapt the behavior of older individuals. The more successful the older individual is, the higher is the probability of adaptation of its strategy by young individuals.

In this ecological model there is a number of strategies and a number of individuals using these strategies. Both values are fixed. At the beginning, all strategies have the same number of individuals playing the strategy. But there are multiple generations and in each generation all individuals compete in a standard *round-robin* tournament. Afterwards, for each strategy their gained payoffs are accumulated. The number of individuals using a strategy in the next generation is proportional to the sum of the payoffs of this strategy in the last generation. The tournament is finished after a fixed number of rounds or if this form of evolution comes to an equilibrium. The strategy with the highest number of individuals wins.

In the ecological tournament, strategies performing badly will die out over time while good strategies get more and more individuals playing it. However, due to the fact that a strategy can only be played by a discrete number of individuals, the number of individuals using a strategy in the next generation cannot be exactly proportional to the sum of the payoffs in last generation. For this reason, we are using an infinite number of individuals and therefore work with fractions of a population instead of discrete numbers. However, the drawback of this approach is that an equilibrium is never reached. For this reason, a tournament will be considered as being over when there are no further remarkable changes in the fractions of population, where remarkable can be substituted to a constant value near zero.

This environment is a good candidate for being able to lower advantages of submitting a master strategy and slave strategies at the same time. Due to the fact that after the first rounds the fraction of population of slaves strategies are heavily reduced, their impact on the further evaluation of the strategies is decreased as well.

C. *The Jerusalem over all Environment*

This tournament environment is named on the game “‘musical chairs””, and sketched also at the same time the underlying idea clearly.

We use again different rounds of single *round-robin* tournaments. The idea is to remove the last strategy after an evaluation of a *round-robin* tournament is performed. Now simply this tournament is accomplished in order to determine the strategy at the last place. Then a new tournament round starts, whereby again the last strategy is removed. This process will continue until only one strategy remains or all strategies are cooperating throughout their games. From now on all strategies receive exact the same score in the last round. In

this way it can be achieved that a slave strategy is punished for its support.

The intention of *Jerusalem over all* is that both types of slaves, *uncooperative slaves* and *cooperative slaves*, are usually found at the bottom places of the standard environment. *Uncooperative slaves* perform still more badly than cooperative, because cooperation is worthwhile generally.

The tournament evaluation remain unchanged from the *round-robin* tournament. All points from the current *round-robin* tournament are immediately added to the overall points of the strategy. After that the strategy with the fewest points is removed from the tournament. Each strategy gets all points she had earned inclusive the round it was removed. In this kind of environment the strategies still get the chance to accumulate points and this points can rescue the strategy through possible following worse rounds.

D. The Jerusalem Environment

In this environment the strategy with the smallest score of the current *round-robin* tournament is removed. The score of each strategy is only summed up if the whole tournament is finished. So the selection is only over the actual round, and in this kind of environment the strategies get no chance to accumulate points. Each slave strategy, which served its master in the current round, will immediately punished. But if a non-master-slave strategies performs bad in one round it can be kicked out too, even if it was on the first place the rounds before.

IV. EXPERIMENTAL SETUP

In this section the experimental setup is presented, which is necessary to reproduce our results. The used set of strategies can be obtained at the web-page: <http://www.ipd-dortmund.de>.

A. Environment Settings

All tournaments are repeated in three passes. For each pass, NR is calculated by equation 1. For this the minimum number of rounds is set to $gl_{min} = 100, 500, 1000$. Due to this, there is a probability of $10/NR$ that the current pass ends. So the probability to finish the current tournament is $P(NR) = 0.1, 0.02$ or 0.01 per round after the actual value of gl_{min} is reached. The overall points of a strategy for a tournament is the sum off all gained points in all tournaments. To hold the effect of random low, all rounds are repeated ten times. Of these ten experiments the average will count. This average is an indicator for the place in the whole game. In case of two strategies which are on the same place, the same rules apply as within the *round-robin* tournament. The other strategies which are affected by this deciding game move down one place.

Both *Jerusalem* environments are simply based on the above PG474 evaluation. In case of the ecological environments two complete *round-robin* tournaments are computed before the ecological tournament starts. This was done in order to reduce noise and computation time. Afterwards, the average score earned by both strategies in each pairing of strategies in the *round-robin* tournament is calculated and used as base for the

ecological tournament. An ecological tournament is considered over when the fractions of population change almost nothing anymore. The number of generations was set to 1200.

B. Strategy set

Within the scope of the Campus fest of the University of Dortmund, at June of 2005, we organized a *round-robin* tournament based on the modified version of the PG474. 64 strategies were sent in for the tournament. Due to the fact that 2 strategies for each participant were allowed, 10 of the strategies are masters and 10 their slaves. This set of strategies represents the basic test set for all tournament environments in the next section. In addition, to ensure a relative representative and realistic test set 17 so called standard strategies listed in table I are added, too. Finally the *RANDOM*-strategy complete our test set of strategies.

TABLE I
STANDARD STRATEGIES OF OUR STRATEGY-SET.

| | | |
|------------------|------------------|---------------|
| Always defect | Always cooperate | Tit-for-tat |
| Tit-for-two-tats | Per kind | Per nasty |
| Go by majority | Spite | Joss |
| Mistrust | Gradual | Prober |
| Pavlov | Tester | Hard majority |
| Slow Tit-for-tat | Hard Tit-for-tat | |

C. Classification of strategies

For the analysis of the results it is necessary to classify strategies. In [Axe84] Axelrod introduced some possibilities to classify strategies. Furthermore in [PdUDB05] advanced possibilities to classify strategies are specified.

Classification of strategies:

- **Master:**

- The master always defects his slave;
- A sequence must be played to recognize a strategy;
- The master has an alternative strategy to play;

- **Slave:**

- The slave always cooperates his master;
- A identification sequence must be played to recognize a strategy;
- The slave has a alternative strategy to play;

- **Friendly:** Does not defect first [Axe84];

- **Aggressive:** Always defects again despite the punishment of the opponent after own defection (at least in between);

- **Forgiving:** Shows willingness to cooperate again after the defection of the opponent [Axe84];

- **Reactive:** Answers defection of the opponent with own defection. After one or more defections of the opponent the strategy plays in finite time a certain, on the number of opponent defections dependent number of defections. The defection does not have to be played in order of time. The count of defection must be at least one time higher as 0.

V. MAIN RESULTS

In this section the results of our experiments are listed, before in the next section an analysis and discussion of the obtained results is done.

A. Results of the tournaments

Table II shows the result of the tournaments in each environment. Due to the lack of space we focus on the ranks of the masters and slaves. As reference, the well-known strategies *TIT FOR TAT* and *RANDOM* are listed, too.

If we look at table II we see that the first two strategies are master strategies. The ranks 3–5 are variants of the standard strategy gradual, often with different ways in calculating the number of defections. The best standard strategy in the PG474 tournament (column *B*) is *GRADUAL* on rank 12. *TIT FOR TAT* only performs on rank 14. In general the slave strategies perform badly. The first slave can be found on rank 59. The other 9 slaves can be found on the last ranks. *RANDOM* leaves 14 strategies behind on rank 70. The first 18 ranks are either master strategies, *TIT-FOR-TAT* variants or *GRADUAL* variants.

The results of the *Jerusalem* and *Jerusalem over all* environments are listed in columns *D* and *E*. But we must see that in both variants the first rank is kept unchanged by the master *Netzer*. The second master however loses many places in both environments. If we continue to look at the table, then we can state that in principle in both environments the masters lose places and this partially in a very large amount. In addition the slave strategies remain in the lower field of the tournaments (ranks 59–82). The same trend can be observed in the ecological environment (column *C*). Here however, the order of the ranks of the slaves varies more than in the *Jerusalem* environments. An exception forms the masters strategy *MNI* in all three cases. Here the strategy goes to the top of the table. If we look at the standard strategies and the non-master-slave strategies, then no general behavior is to be determined. While in the *Jerusalem* environments a general upward trend is to be observed, the behavior in the ecological environment is indifferent.

VI. ANALYSIS OF THE RESULTS

In the last section we have shown our main results from all tournaments. The observation of the obtained results lead to the assumption that the new environments are able to reduce the influence of master-slave strategies. In this section we will analyze whether this assumption is correct or not. At first we will analyze the common grounds, before we come to their differences.

A. Common points of all new environments

Both *Jerusalem* environments as well as the PG474 *round-robin* and *ecological* environment belongs to the class of monotonic environments, because we simply add positive scores. Based on our negative result in section II-C it is not possible in all this environments that a strategy without a slave performs better than its identical master strategy. It

TABLE II

RESULTS OF ALL TOURNAMENTS: STARTING FROM THE RESULTS OBTAINED FROM PG474 (COLUMN *B*), THE ABERRATIONS FROM THIS RANKS USING THE OTHER ENVIRONMENTS ARE LISTED IN COLUMN *C–E*. + $-c$ MEANS, THAT THIS STRATEGY PERFORMS c RANKS BETTER OR WORSE THAN IN THE PG474 TOURNAMENT. COLUMN *C* DENOTES TO THE ECOLOGICAL TOURNAMENT, COLUMN *D* TO *Jerusalem* AND COLUMN *E* TO *Jerusalem over all*. COLUMN *A* HOLDS THE NAMES AND THEIR CLASSIFICATIONS.

| A | B | C | D | E |
|--|-----|-----|-----|-----|
| (Master) <i>Netzer</i> | 1 | 0 | 0 | 0 |
| (Master) <i>Meister</i> | 2 | -3 | -10 | -9 |
| (Gradual variant) <i>ThmodGradual2</i> | 3 | -4 | -4 | -10 |
| (Gradual variant) <i>ThmodGradual1</i> | 4 | +2 | +2 | +2 |
| (Gradual variant) <i>Simon2</i> | 5 | -6 | +2 | +2 |
| (Master) <i>MNI</i> | 6 | +3 | +2 | +2 |
| (TFT variant) <i>Sebrassi IPD II</i> | 7 | +3 | +2 | +2 |
| (Gradual variant) <i>Schwalli Faku</i> | 8 | -5 | 0 | +1 |
| (Gradual variant) <i>Simon</i> | 9 | 0 | +3 | +3 |
| (standard strategy) <i>Gradual</i> | 12 | -3 | +1 | +3 |
| (standard strategy) <i>Tit for Tat</i> | 14 | -5 | 0 | -1 |
| (Master) <i>Zweiundvierzig</i> | 20 | -21 | -13 | -23 |
| (Master) <i>He-Man</i> | 26 | -17 | -8 | -19 |
| (Master) <i>Young Capone</i> | 40 | -14 | -10 | -14 |
| (Master) <i>Raskolnikoff</i> | 48 | -6 | -4 | -6 |
| (Master) <i>Green Shark</i> | 50 | -3 | 0 | -1 |
| (Master) <i>Major Eisberg</i> | 52 | -6 | -2 | -4 |
| (Master) <i>Throat</i> | 53 | -6 | -4 | -5 |
| ... | ... | ... | ... | ... |
| (Slave) <i>MNI</i> | 59 | +2 | 0 | 0 |
| (standard strategy) <i>Joss</i> | 66 | -3 | -2 | -2 |
| (standard strategy) <i>Random</i> | 70 | +2 | +1 | +1 |
| (Slave) <i>Zweiundvierzig</i> | 71 | -2 | -1 | -1 |
| (Slave) <i>He-Man</i> | 72 | -3 | -1 | -2 |
| (Tester variant) <i>Resistance is futile</i> | 73 | +1 | +2 | +2 |
| (Slave) <i>Green Shark</i> | 74 | 0 | 0 | -1 |
| (Slave) <i>Young Capone</i> | 75 | -2 | 0 | -1 |
| (standard strategy) <i>Per nasty</i> | 76 | +5 | 0 | +3 |
| (Slave) <i>Raskolnikoff</i> | 77 | -1 | 0 | 0 |
| (Slave) <i>Netzer</i> | 78 | -4 | 0 | 0 |
| (Slave) <i>Throat</i> | 79 | -1 | 0 | 0 |
| (Slave) <i>Major Eisberg</i> | 80 | +4 | 0 | 0 |
| (Slave) <i>Meister</i> | 81 | +2 | 0 | 0 |
| (standard strategy) <i>ALLD</i> | 82 | +1 | 0 | 0 |

appears that the masters *NETZER*, *MEISTER* and *MNI*, play a form of either *TIT FOR TAT* or *GRADUAL* in the case of not recognizing their slave. But *TIT FOR TAT* as well as *GRADUAL* itself were incorporated in this tournament as standard strategies. So we can examine the negative result for this three masters and verify that all of them perform better (ranks 1, 2 and 6) than this two strategies (ranks 14 and 12)(table II). All strategies are classified to either master or slave or the second strategy they are playing.

Furthermore it can be seen that all slaves are usually at the bottom ranks in all environments. The ranks of these slaves are hardly changing in our new environments, so the robustness is very high. The slave *MNI* is a *cooperative slaves* and thus remains much longer in tournament than the other slaves. Instead all *uncooperative slaves* can be found at positions 71–81.

Strategies which are *friendly* and *reactive* are very success-

ful in the game. It is obvious that *NETZER* performs very good, so this means that he seem to have a *friendly* and reactive second strategy. Indeed his second strategy consists of a complex mixture of *TIT FOR TAT* and *GRADUAL* with recognizing certain strategies. He is the only one out of 10 masters, who is not alternated in the different environments at all, but as we have showed in our negative result that in some cases we cannot avoid that.

It is obviously that as well in both *Jerusalem* variants as in the PG474 tournament the lowest strategy is always *ALWAYS DEFECT*. In the *ecological* environment all slaves except one are better than *ALWAYS DEFECT*, too. Consider the *uncooperative slaves*: They will in most cases (81 out of 82 if identification patterns do not overlap) defect everything like *ALWAYS DEFECT* and in addition perform poorer than mutual defection against their master. Nevertheless the slaves are better, which is not a surprising result. The reason is that the slaves have to play an identification pattern in order to properly recognize their master. Usually the length of this pattern in this tournament was 5-20 moves with approximately a half cooperates and a half defections. With this cooperation moves the slaves are getting more points than they are loosing by their master. Note that this 5-20 moves have to be played against all 82 strategies oppositional to one single game against their master. This is why *ALWAYS DEFECT* is on the last rank instead of *uncooperative slaves*.

B. Jerusalem tournaments

To demonstrate the effect of our two new *Jerusalem* variants we will first evaluate a tournament with 5 strategies, but beside this with exactly the same experimental setup as the PG474 tournament. We used a reduced strategy-set *s1*, consisting of *TIT FOR TAT*, *ALWAYS COOPERATE*, *ALWAYS DEFECT* and a master called *MEISTER* with his associated *uncooperative* slave *SLAVE MEISTER*. This master is based on *TIT FOR TAT*, so we can once again use our negative result in II-C and show that *MEISTER* beats *TIT FOR TAT*. The accumulated points in the *Jerusalem* variant of this small tournament can be seen in figure 1. Here, the slave is removed after the first round, but nevertheless he could give his master an advantage in that first round, which the master maintains. In figure 2 the same strategy set in the *Jerusalem over all* environment is shown. Note that in this variant always the strategy with the smallest over all score is removed after each round. Though all strategies are cooperating after the second round, the tournament is played until only one strategy is left. The reason is that strategies can collect points for further round in this variant and therefore should be estimated different. In the last round there is only the master left, but he gets points, though, because every strategy has to play against itself.

Now we will return to the whole tournament of 82 strategies. *Jerusalem* uses the fact of eliminating the poorest performing strategies after each *round-robin* round. Therefore the slaves are eliminated early in tournament, what leads to an punishment of the masters. That is exactly what we want for fair tournaments.

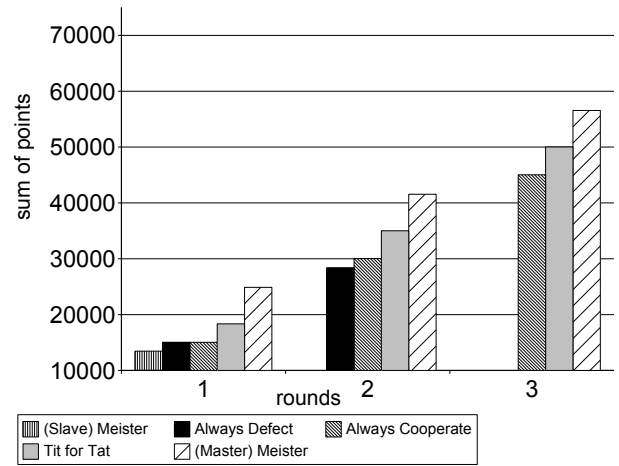


Fig. 1. 5 strategies in the *Jerusalem* environment with $NR = 115, 517, 1035$ is depicted. Here, *MEISTER* gets an unavoidable boost in the first round because of his slave *SLAVE MEISTER*. This poor performing slave is sorted out in this very first round, so from this round on, the master will not gain points from him. As the master is based on *TIT FOR TAT*, he is not able to outperform further.

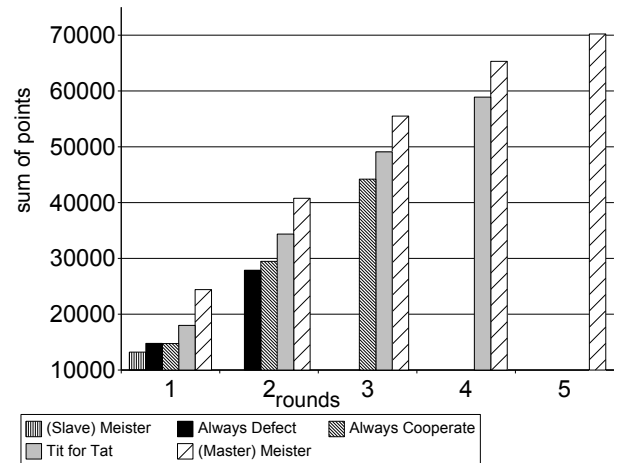


Fig. 2. The *Jerusalem over all* environment: The same behavior as in *Jerusalem* can be observed. But here, however cooperation takes place after the second round.

As already stated the slave *MNI* is an *cooperative slaves*. Here the method of *Jerusalem* and *Jerusalem over all* is not as promising as with *uncooperative slaves*, because *MNI* is able to boost his master for more rounds, since he is removed later (table II). That is the reason why his master is the only one who is able to perform better than in the standard *round-robin* environment. All 8 other masters except the winner *NETZER*, who remains constant, are performing poorer in both *Jerusalem* environments than in the *round-robin* environment.

Naturally the strategy set changes over rounds. Starting with the original strategy set the poorest performing strategy is removed and since this will usually be a mostly defecting strategy, the tournament strategy set changes to a more cooperative behavior. This lead to a situation where all left strategies are cooperating. Therefore it is clear, that

aggressive strategies like slaves are kicked out very early. Since cooperative strategies make more points, the differences in the score between strategies on top ranks is relative high. Because of this and the removing process, the robustness will usually be higher on lower ranks, but we will show that this does not matter:

Even without excluding master-slave strategies, over 80% of the strategies in the *Jerusalem* environment had a standard position derivation of less than 1. In the *Jerusalem over all* environment there were over 75% with a standard position derivation of less than 1. In the first case the standard position derivation was on an average of 0,717, in the latter environment an average of 0,661. This good robustness led to nearly the same ranks in all 10 repetitions.

Figures 3 and 4 refer to the progression of several selected strategies. They consist of the masters *YOUNG CAPONE* and *NETZER*, the slave *SLAVE YOUNG CAPONE*, and *RANDOM* and *ALWAYS COOPERATE*. Note that while *ALWAYS COOPERATE* performs better than master *YOUNG CAPONE* in the *Jerusalem* environment, this changes in the *Jerusalem over all* environment. *YOUNG CAPONE* raises higher, but was thrown out before *ALWAYS COOPERATE* in the *Jerusalem* environment. This rank-flipping can be explained through the possibility to save advantages over rounds in the *Jerusalem over all* environment. *YOUNG CAPONE* is therefore able to use his head start to remain longer in tournament. *ALWAYS COOPERATE* doesn't even reach the round where all strategies are cooperating, because here it is more difficult to walk over the barrier of mostly defecting strategies. The concave lines of the better performing strategies refer to the drowning strategy pool at later rounds in evaluation. As well as in the examples above the *Jerusalem over all* environment is finished when there is only one strategy left.

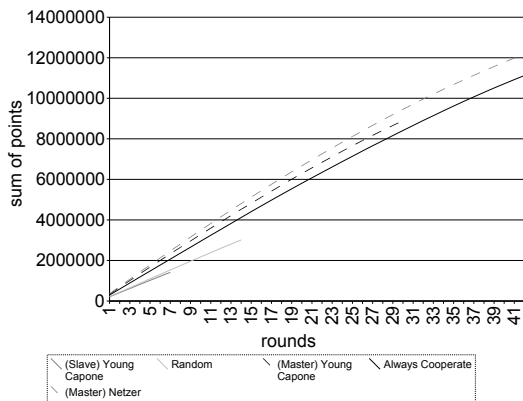


Fig. 3. Progression of selected strategies in the environment *Jerusalem*. It can be seen, that the slave *SLAVE YOUNG CAPONE* is removed very soon after the seventh round, while his master lasts to the 30th round. Further the lines are all slightly concave because in following rounds there will be less strategies left, resulting in less points for every strategy involved. In round 41 all strategies are cooperating, so tournament is finished here.

C. Ecological environment

Based on *s1* we made our first evaluation with the ecological environment. The results are depicted in figure 5.

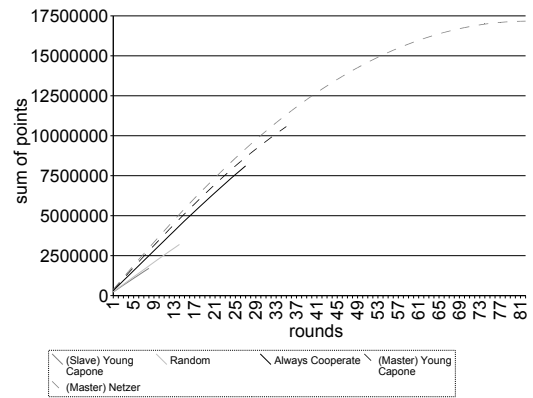


Fig. 4. Progression of selected strategies in the environment *Jerusalem over all*. In this variant all rounds are played, so that only one strategy is left at the end. There is an eye-catching concave line of *NETZER*. This is representative for all strategies that run nearly until end of tournament, because the number of left strategies decreases linear and so do the points. Beside of this all strategies have a nearly linear behavior, except a slightly loss of points of the master *YOUNG CAPONE* beginning at round 25, which could result of the changing environment.

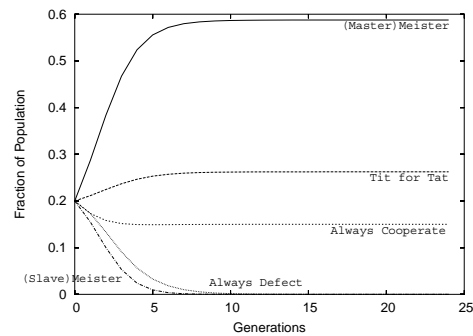


Fig. 5. Ecological environment with $NR = 126,515,1115$, and over 25 generations. After 10 generations the master based on *TIT FOR TAT* and *TIT FOR TAT* have nearly the same changing rate in population.

Analog to *Jerusalem* and *Jerusalem over all* the master *MEISTER* gets an unavoidable boost in the first generations from his slave and wins the tournament with the highest fraction of the population. The master strategy gets the boost from his slave in the first 7 generations and has nearly three times the population of the identical *TIT-FOR-TAT* strategy without a slave. In this small tournament the advantage of the master strategy is much higher than in the tournament with all 82 strategies, because it does only need a few generations to come to the end of the tournament. That means if there are only few strategies in the tournament, the advantage of a master strategy is higher, because there are less generations. The slave strategy performs very badly and its fraction of population is approaching zero after 7 generations, so that it cannot support the master strategy any more (although the slave never really dies out at all, it does not deliver much points for the master). After *ALLD* and slave *SLAVE MEISTER* have lost most of their fraction of population, there is mostly cooperation in the remaining population and no furthermore notable changing in the fraction of population.

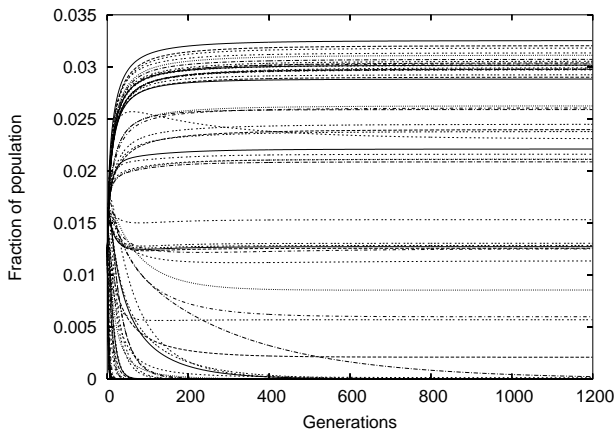


Fig. 6. 82 strategies in the ecological tournament over 1200 generations. The slave strategies die out in the first generations. After 1200 generations there are no remarkable changes in the fractions of population. Due to the number of strategies the advantage of the master strategies is much lower in $s1$.

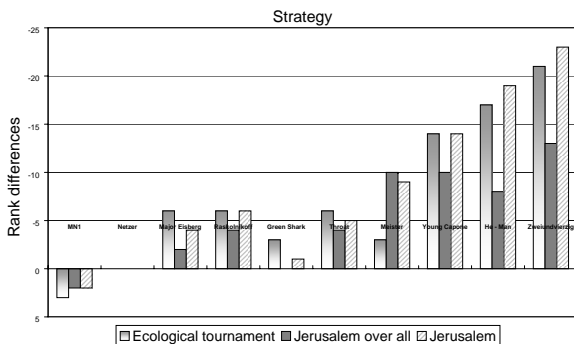


Fig. 7. The rank differences of all 10 master strategies in our different environments compared with the *round-robin* tournament. All environments lower the advantage of the master strategies. Only one strategy (*MNI*) performs better. *Jerusalem* shows the highest effect on reducing the advantage of master-slave-strategies.

We evaluated a ecological tournament with all our 82 strategies, which is illustrated in figure 6. The Figure 7 shows that it lowers the advantage of all master strategies except of (Master) *MNI*. After 1200 generations there are no remarkable changes in the fraction of population, due to the fact that almost all strategies are cooperating. In this tournament we need much more generations to reach the end of the tournament. Here we need round about 1200 generations. Therefore the slave strategies loose most of their population in the first generations and cannot supply their master strategy any longer with additional points.

There are some differences in the order of the slave strategies, thus they are also remaining on the last ranks. This is a difference compared to the *Jerusalem* environments. This difference is based on the fact that in *Jerusalem* the strategies are removed from the tournament and in the ecological tournament they only loose their fraction of the population. For example *ALLD* gains one rank because he it profits longer from the growing cooperative strategies. In a *Jerusalem* environment

instead *ALLD* is directly removed at the beginning, because all slaves perform better in the beginning because of their identification pattern. The ecological environment shows also that it has a good robustness. The standard position derivation of strategies was over 50% less than 1 and over 80 less 2. On average it was 0,717.

VII. CONCLUSIONS

In this paper the problem was treated whether it is possible to create a fair tournament environment in order to solve the problem of master-slave strategies. We showed that there is no practical way to prevent collaboration between masters and their slaves through organisational measures nor through an automatic way of recognizing identification pattern. Within the class of *monotonic* environments, and all well-known environments belong to it, it turned out that it is not possible to eliminate this collaboration from scratch. Nevertheless we showed that it can be significant reduced instead. For this purpose we introduced four new environments, namely PG474, *Jerusalem*, *Jerusalem over all* and a modified ecological environment. It occurred that especially *Jerusalem* is able to lower the advantage of master-slave strategies and therefore establishes a fair way of hosting tournaments, where master-slave strategies are not allowed. The best results can be achieved on masters with *uncooperative slaves*, because they are in general at the bottom ranks, leading to a punishment of their masters in the ecological and the *Jerusalem* environments.

It would be an interesting question to expand this results for noise- or multiplayer IPD-games. Our future work includes the examination of improvements with *cooperative slaves*, since their detection is more difficult. Furthermore all results could perhaps be extended in a non-monotonic environment, but as already showed this seems to contradict common concepts of tournaments.

REFERENCES

- [Axe84] R. Axelrod. *The Evolution of Cooperation*. Basic Book Inc., New York, 1984.
- [DM95] J. P. Delahaye and P. Mathieu. Complex Strategies in the Iterated Prisoner's Dilemma. In A. Albert, editor, *Chaos and Society*, volume 29 of *Frontiers in Artificial Intelligence and Applications*, pages 283–292. Université du Québec à Hull, Canada, IOS Press/Presses de l'Université du Québec, 1995.
- [FOG93] DB FOGEL. Evolving Behaviors in the Iterated Prisoner's Dilemma. *Evolutionary Computation*, 1:77–99, 1993.
- [HC88] J HIRSHLEIFER and JM Coll. What Strategies Can Support the Evolutionary Emergence of Cooperation? *Journal of Conflict Resolution*, 32(2):367–398, 1988.
- [MUE88] U MUELLER. Optimal Retaliation for Optimal Cooperation. *Journal of Conflict Resolution*, 31:692–724, 1988.
- [NAC92] JH NACHBAR. Evolution in the Finitely Repeated Prisoner's Dilemma. *Journal of Economic Behavior and Organization*, 19:307–326, 1992.
- [PdUDB05] Kajendran Kosslers Polaszek Reimholz Schmidt Seis Slodzinski Steeg Wiemann Zimmermann Projektgruppe474 der Uni Dortmund : Baranski, Ehlers. Zwischenbericht der PG 474: Neue Anstze fr das Gefangenendilemma, Novemeber 2005.
- [PM06] Bruno Beauflis Philippe Mathieup, Jean Jean-Paul Delahayed. <http://www.lifl.fr/ipd/misc/cec2004.html>, January 2006.

B. High-Order Punishment and the Evolution of Cooperation

High-Order Punishment and the Evolution of Cooperation

Track: Coevolution

ABSTRACT

The Prisoner's Dilemma and the Public Goods Game are models to study mechanisms leading to the evolution of cooperation. From a simplified rational and egoistic perspective there should be no altruistic cooperation in these games at all. Previous studies observed circumstances under which cooperation can emerge. The effect of viscosity, i.e., spatially distributed individuals with limited mobility and local interactions, on the evolution of cooperation will be discussed. Furthermore we will analyze the existence of altruistic punishment in humans. This paper demonstrates that high-order punishment opportunities can maintain a higher cooperation level in an agent based simulation of the evolution of cooperation.

Categories and Subject Descriptors

I.2.1 [Computing Methodologies]: Artificial Intelligence Applications and Expert Systems[Games]; I.2.11 [Computing Methodologies]: Artificial Intelligence—*Distributed Artificial Intelligence, Intelligent agents*

General Terms

Coevolution

Keywords

IPD, Public Goods Game, Cooperation, Punishment

1. INTRODUCTION

Altruistic cooperation is common in human nature even though from a simplified and shortsighted "rational" perspective there should be no cooperation at all. The *Prisoner's Dilemma* (PD) and the *Public Goods Game* (PGG) are often used models to study mechanisms that lead to the evolution of cooperation.

The PD is a classical non-zero-sum game of game theory. It can be formulated as follows:

Two men (agents), charged with a joint violation of law, are held separately by the police. Each is told that (1) if one confesses and the other does not, the

former will be given a reward. . . and the latter will be fined (2) if both confess, each will be fined. . . At the same time, each has a good reason to believe that (3) if neither confesses, both will go clear [1].

The payoff of each agent depends on the decision made by these two agents. Table 1 provides the score in each case. If agent *A* and *B* confess, both serve a penalty of $R = 3$ years. If *A* confesses and *B* defects, *A* serves a penalty of $S = 0$ years and *B* serves a penalty of $T = 5$ years. If *A* and *B* defect, both serve a penalty of $P = 1$ year. If both agents only care about minimizing their own jail term, it is beneficial for each agent to defect, no matter how the other agent behaves. The following inequalities describe the resulting prisoner's dilemma: $T > R > P > S$.

The PGG is well known in experimental economics. In the basic game subjects choose how many of their income to put in a public pool. Each subject keeps the token not contributed plus an even split of the tokens in the public pool. No matter what the other subjects do, it is beneficial for each suspect not to contribute. The PGG could be appreciate as a PD with changed payoff matrix (e.g., $T = 5$, $R = 3$, $P = 1$, and $S = -1$).

In these games it is assumed that cooperation is costly to the individual and only opponents can benefit from it. The so-called *free-rider problem* occurs, if it is beneficial for the individual not to cooperate, no matter how the opponent (or the society) reacts. Free-riders are actors who consume more than their fair share of a resources. The Stanford Encyclopedia of Philosophy describes the practical relevance of the free-rider problem for everyday life as follows:

Example 1 For example, if each of us pollutes less by paying a bit extra for our cars, we all benefit from the reduction of harmful gases in the air we breathe[. . .] If all of us or some subgroup of us prefer the state of affairs in which we each pay this bit over the state of affairs in which we do not, then the provision of cleaner air is a collective good for us[. . .] Unfortunately, my polluting less does not matter enough for anyone—especially me—to notice. Therefore, I

Table 1: Payoff matrix of the classical PD showing the served penalty (e.g., $T = 5$ years, $R = 3$ years, $P = 1$ year and $S = 0$ years)

| | <i>B</i> confesses | <i>B</i> defects |
|--------------------|--------------------|------------------|
| <i>A</i> confesses | R / R | T / S |
| <i>A</i> defects | S / T | P / P |

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CEC '06 Seattle, Washington USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

may not contribute my share toward not fouling the atmosphere. I may be a freerider on the beneficial actions of others [10]. \square

Many research had been done to describe the circumstances under which cooperation can emerge despite of these drawbacks. The theory of *kin-selection* focus on cooperation among individuals that are in some ways closely related. *Spatial structure* is a factor that can sustain cooperation in a small range of PD payoff parameter [8, 9, 4]. The theories of *direct reciprocity* focus on bilateral long-term interactions [4]. The theories of *indirect reciprocity* explain how cooperation can emerge through *reputation* which directly leads to *reward* and *punishment* mechanism [11, 4].

In the remainder of this paper we consider the tendency of individuals to punish free riders. In most cases of punishment, sending a fine has a cost to the opponent and to the punisher itself. Particularly with regard to the cost to the punisher itself, it is unclear why humans should engage in costly punishment. This leads to the so-called *second-order free-rider problem*.

This paper combines the agent-based model of Hammond and Axelrod [6] with punishment opportunities. We examine the effect of punishment opportunities on the evolution of cooperation and present some new results to the question “Can punishment opportunities enforce the evolution of cooperation?” Figure 1 depicts a typical run of the extended model.

Software and data from this article can be downloaded from [12].

The remainder of this paper is arranged as follows. Section 2 describes the agent-based model of Hammond and Axelrod, the punishment model of Fehr and Gächter [5], and the concept of high-order punishment [2, 7]. Our extended agent-based model is introduced in Sect. 3. Section 4 presents and discusses the simulation results. A conclusion and an outlook are given in Sect. 5.

2. BACKGROUND

This section provides a brief overview of Hammond’s and Axelrod’s agent-based models. Furthermore, existing experiments concerning the willingness of humans to punish free riders are presented.

2.1 The Effect of Cost-Benefit Ratio on the Evolution of Cooperation

Hammond and Axelrod presented three agent-based models to study minimal conditions for the evolution of cooperation: (i) the *Null Model*, (ii) the *Viscosity Model*, and (iii) the *Tag Model*. These models do not memorize previous interactions, communication of reputations, leadership, etc.

2.1.1 Null Model

In the *Null Model* (M_{null}) each agent is either an altruist who always cooperates or an egoist who never cooperates. There is no structure in the agents environment. Each time period of the model consists of four stages: immigration, interaction, reproduction, and death.

1. *Immigration*. The model starts with an empty population. In the immigration stage a single agent with a random strategy enters the population. At the beginning of each period each agent has the same initial value of 12% as its *potential to reproduce* (PTR).
2. *Interaction*. As mentioned above, this model imposes no structure on the environment, so each agent is paired randomly with another agent during the interaction phase.

3. *Reproduction*. Cooperative agents give help and pay a *cost* C , i.e., a decrease in the agents PTR for each donation. Defective agents do not give help. Receiving help has a *benefit* B , i.e., an increase in the agents PTR. The *cost-benefit ratio* C/B (CBR) serves as a guideline for the analysis. Interaction can be understood as a one-move Prisoner’s Dilemma. In the reproduction stage agents are randomly chosen and reproduced with a probability equal to their own PTR. While the maximum number of agents, e.g. 2500, is not exceeded, new offspring is generated. Offspring inherit the strategy of their parents, with a mutation rate of 0.5%.

4. *Death*. Finally, there is a 10% risk of dying for each offspring, making room for future immigrants or offspring, in the dead stage.

The outcome of the M_{null} model therefore is that no strategies based on kinship or reciprocity are produced. As expected, under these conditions there is no emergence of cooperation. In a benign environment (in terms of a low cost-benefit ratio i.e. $C = 1\%$ and $B = 3\%$) the frequency of cooperation is about 4.6%.

2.1.2 Viscosity Model

The *Viscosity Model* (M_{visc}) is a simple extension of the M_{null} model. Viscosity means that the members of a population are spatially distributed, have limited mobility and interact locally. So the agents are now distributed on a grid (50×50 cells by default). At the immigration stage the agent is placed randomly on the grid. The grid has wrap-around borders so that each cell has exactly four adjacent cells. At the interaction stage each pair of adjacent agents interact in the same one-move Prisoner’s Dilemma as in the M_{null} model. Mathematically speaking, two agents are adjacent, if their L_1 distance (city block distance) is one. At the reproduction stage, offspring is placed on the adjacent cell if it is not occupied.

The simulation results show that viscosity sustains the evolution of cooperation in a sufficiently benign environment (in terms of a low cost-benefit ratio, e.g., $C = 1\%$ and $B = 3\%$). It does not maintain the evolution of cooperation in an austere environment (in terms of a higher cost-benefit ratio, e.g., $C = 2\%$ and $B = 3\%$). See Fig. 2 and also [6].

2.1.3 Tag Model

The *Tag Model* (M_{tag}) is a simple extension of the M_{visc} model. In addition to the features from above, each agent has a tag (one of four colors) which specifies its group membership. The agent’s strategy depends on the opponent’s color. A possible strategy might be cooperating while interacting with members of its group and defecting otherwise.

The simulation results show that in the M_{tag} model there is no emergence of cooperation. In a benign environment (in terms of a low cost-benefit ratio, i.e., $C = 1\%$ and $B = 3\%$) the frequency of cooperation is about 22.6%.

Figure 2 illustrates that combining the M_{visc} and the M_{tag} model allows a given level of cooperation to be maintained even if cooperation is quite costly (in terms of a low cost-benefit ratio, i.e., $C = 2\%$ and $B = 3\%$).

2.2 First-Order Punishment in Humans

Fehr and Gächter [5] demonstrated experimentally that punishment opportunities generate a higher *average contribution level* in public goods games. The average contribution level is defined as the mean contribution of all participants during all as well as final periods in their public goods games. In this experiment reciprocity has no influence on the contribution level. Fehr and Gächter also

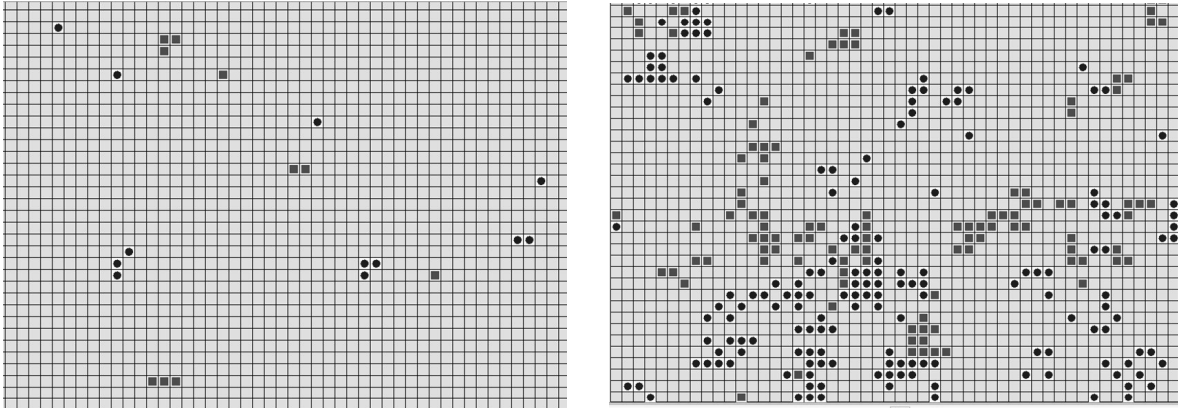


Figure 1: A typical run of the model after 35 periods (*left*) and 350 periods (*right*). In the early periods the scattered immigrants create regions of similar agents. If the grid becomes more and more occupied the different regions become adjacent to each other. The black circle on a cell represents an altruist and the dark gray rectangle an egoist

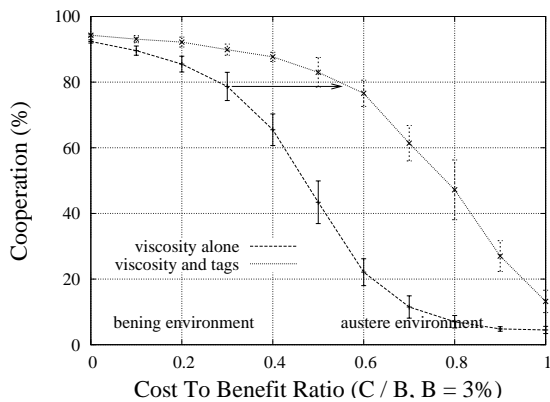


Figure 2: Combining the M_{visc} model (spatially distributed population, limited mobility, local interaction) and the M_{tag} model (strategy depending on the co-player's group membership) allows a given level of cooperation to be maintained in more austere environments

observe a widespread willingness of human cooperators to punish free riders even if punishment is costly and does not provide any benefit for the punisher.

The experimental model considered by Fehr and Gächter consists of a public goods game with n human players. There is a *Stranger Treatment* and a *Partner Treatment*, each equipped with and without punishment opportunities. In the Partner Treatment the same group of $n = 4$ subjects plays a finitely repeated public good game for ten periods. The group composition does not change between different periods. In the Stranger Treatment the total number of participants is randomly partitioned into smaller groups of size $n = 4$ in each of the ten periods. The group composition in the Stranger Treatment changes randomly from period to period.

In each period each of the n subjects receives an endowment of y tokens. At the *Contributing Stage* an individual can either keep these tokens or contribute g_i tokens ($0 \leq g_i \leq y$) into the public pool. Each of the n individuals keeps the tokens they do not contribute and receives an even split of the tokens from the public pool. The group as a whole does best when everyone contributes all of their tokens into the public pool. But everyone, regardless of con-

tributing or not, benefits from the tokens in the public pool. That implies that under rational and egoistic assumptions full free-riding ($g_i = 0$) is a dominant strategy at least in the Stranger Treatment.

Fehr and Gächter extended the simple public good game by introducing a second decision stage. In the *Punishment Stage* individuals have the opportunity to simultaneously punish all non-contributing co-players. Group member j can punish non-contributing group member i by assigning so-called punishment points p_j^i to i . For each punishment point assigned to i the first stage payoff of i is reduced by a fine $\beta = 10\%$. For each punishment point p_j^i the first stage payoff of the punisher j himself is reduced by a cost γ . The more punishment points assigned, the greater γ value is.

Fehr and Gächter reported, that punishment opportunities cause a large rise in the average contribution level in the Stranger Treatment as well as in the Partner Treatment. In the no-punishment condition of the Stranger Treatment as well as in the Partner Treatment the average contribution converges to full free-riding over time.

2.3 Second- and Higher-Order Punishment in Humans

Punishment is costly and does not provide any direct benefit for the punisher. Contributing group members who do not punish non-contributing group members are better off than those who punish, because the non-punishing group members do not pay the cost of punishing. Therefore the provision of punishment is itself a public good that is affected from the free-riding problem. The problem of cooperation ended in the problem of second-order free riders.

Example 2 We give a simple example from everyday life for second order free-riders. If somebody is observing a crime without intervening (helping the victim, calling the police), he/she can be punished by law. \square

This may require the opportunity to punish the non-punishing group members (so called second-order punishment). However, this may lead to an endless regress: second-order punishment must be displaced by a third-order punishment, etc., see also the discussion in [2, 7].

3. THE EXTENDED MODEL

We extended the M_{visc} and the M_{null} model with high-order punishment opportunities to study possible effects on the evolution of

cooperation. This model will be referred to as the *extended model* (M_{ext}). We analyze the effect of different *cost-fine ratios* (CFR) on our model and the effects caused by increasing costs of punishment.

The agents in the original agent-based model of Hammond and Axelrod used simple strategies: Each agent A has a *cooperation strategy*, i.e., A is either an altruist who always cooperates or an egoist who never cooperates. In our model we added the ability to punish egoistic and non-punishing agents: the *punishment strategy*.

Immigration, reproduction and death stage in the M_{ext} model are implemented as in the M_{visc} and the M_{null} model. The interaction stage in the M_{ext} model is divided into the *cooperation stage* and the *punishment stage*:

- *Cooperation stage.* At the cooperation stage each pair of adjacent agents interacts in a one-move Prisoner's Dilemma in which each agent chooses whether or not to help the other agent (playing the cooperation strategy). Cooperative agents give help and pay a cost C (denotes a decrease in the agent's PTR) for each donation. Defective agents do not give help. Receiving help has a benefit B (denotes an increase in the agent's PTR).
- *Punishment stage.* At the punishment stage each agent chooses whether or not to punish egoistic or non-punishing co-players from the cooperative stage (playing the punishment strategy). If the simulation's global *punishment level* θ , is set to 0, there is no opportunity to punish other co-players. The interaction stage is finished. If $\theta = 1$, agents have the opportunity to punish egoistic opponents. If $\theta = 2$, agents may punish egoist opponents and directly afterwards punish non-punishing opponents. If $\theta = 3$, each agent has the opportunity to punish an egoist, a non-punishing opponent, and to punish a non-punishing non-punisher opponent.

Punishing agents give a fine to the co-player and pay a *cost* P (denotes a decrease in the agent's PTR) for each atomic punishment action. Receiving a fine has a *cost* F (denotes a decrease in the agent's PTR).

4. RESULTS AND ANALYSIS

Our models use the standard parameters from Hammond and Axelrod [6] unless otherwise stated. Each simulation is run 2000 periods and replicated 10 times. Data are averaged over the last 100 periods. The range shown is plus/minus one standard derivation unit.

4.1 Punishment Enhances the Evolution of Cooperation

At first we focus on our extension of the M_{visc} model. The cost of punishment for the punisher itself P , and the cost of punishment to the opponent F , were both set to 1% in this simulation. This gives a CFR of 1.0. The simulation results are shown in Table 2, a graphical presentation is shown in Fig. 3.

The simulation results indicate that the opportunity of punishment enhances the evolution of cooperation in a more austere environment. An environment is called *austere* when the cost of helping is comparatively high (in terms of a higher cost-benefit ratio, e.g., $C = 2\%$ and $B = 3\%$).

Without punishment opportunities ($\theta = 0$) there is a total of 80% cooperative agents if $\text{CBR} = 0.4$, i.e., $C = 1.2\%$ and $B = 3.0\%$. This level of cooperation can be sustained in more austere environments (i.e., $\text{CBR} = 0.5$) when the first-order punishment opportunity is given ($\theta = 1$). Regarding the higher-order punishment opportunities ($\theta = 3$), cooperation can be sustained in a much

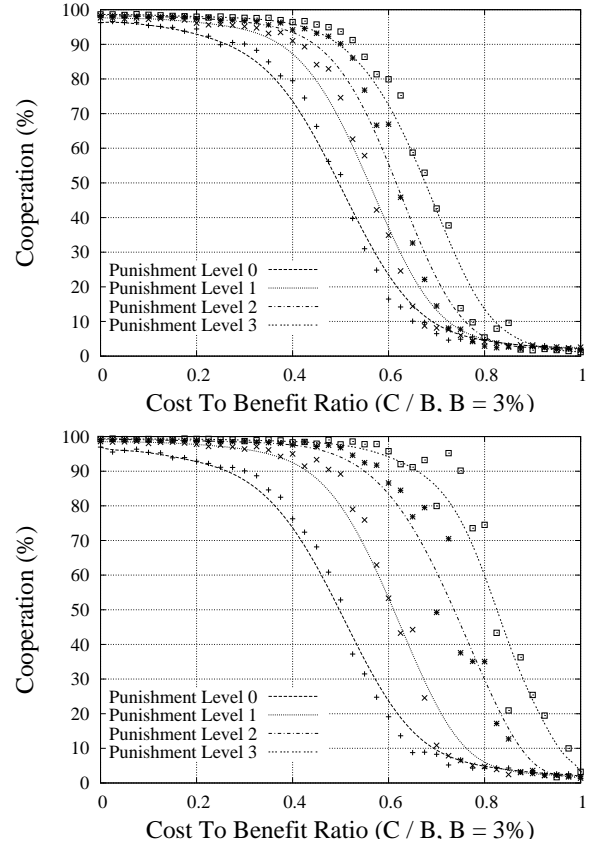


Figure 3: Higher punishment order increases the cooperation frequency. Combining the M_{visc} model (spatially distributed population, limited mobility, local interaction) and the M_{ext} model (opportunity to punish the co-player after being informed about the opponents strategy) allows a given level of cooperation to be maintained in more austere environments. P is the cost of punishment for the punisher and F is the cost for being punished. Left: $P = 1.0\%$, $F = 1.0\%$, right: $P = 3.0\%$, $F = 3.0\%$. Plus, cross, star, and square symbols depict results from punishment level $\theta = 0, 1, 2$, and 3 , respectively. Bezier curves were added to illustrate the data trend

more austere environment. There is still a total of 80% cooperative agents when the CBR is 0.6, i.e., $C = 1.8\%$ and $B = 3.0\%$. These values are shown in boldface in Table 2.

In the next simulation the CFR value was not modified, but the cost of punishment to the punisher itself is set to $P = 3\%$ and the cost of punishment to the opponent is set to $F = 3\%$. The new simulation results support assumptions from previous observations, because the punishment opportunities still enhance the evolution of cooperation in a more austere environment. Higher punishment costs in this simulation—both for the punisher himself and his opponent—actually enhance the evolution of cooperation more drastically. Simulation results are shown in Table 2 and depicted in Fig. 3.

These simulations illustrate why cooperative behavior occurs in more austere environments. In the early periods of a simulation run, immigrants that are scattered all over the grid, create regions of similar agents. If the grid becomes more and more occupied the dynamics are governed by interactions at the borders between regions occupied by altruists and egoistic, see Fig. 1. Examina-

Table 2: Adding punishment opportunities to viscosity allows a given level of cooperation to be maintained in more austere environments when the cost of punishment for the punisher (P) and the cost for being punished (F) are identical. Higher punishment order increases the cooperation frequency (CO). C/B is the cost of giving help (C) to benefit of receiving help (B) ratio. Each case is a 2000 period run and is replicated 10 times. Data are averaged over the last 100 periods. The range shown is plus or minus the standard derivation. Numbers explicit referenced from the text (Section IV) are marked boldfaced

| | | $P = 1.0\%, F = 1.0\%$ | | | | $P = 3.0\%, F = 3.0\%$ | | |
|-------|---------------------|------------------------|--------------|---------------------|--------------|------------------------|---------------|--------------|
| | | $\theta = 0$ | $\theta = 1$ | $\theta = 2$ | $\theta = 3$ | $\theta = 1$ | $\theta = 2$ | $\theta = 3$ |
| C/B | CO (%) | CO (%) | CO (%) | CO (%) | CO (%) | CO (%) | CO (%) | CO (%) |
| 0.0 | 96.26 ± 0.36 | 97.66 ± 0.17 | 98.34 ± 0.19 | 98.63 ± 0.13 | 98.56 ± 0.14 | 99.12 ± 0.10 | 99.29 ± 0.06 | |
| 0.1 | 95.43 ± 0.37 | 97.23 ± 0.26 | 98.13 ± 0.13 | 98.52 ± 0.12 | 98.04 ± 0.13 | 98.99 ± 0.10 | 99.30 ± 0.09 | |
| 0.2 | 94.42 ± 0.32 | 96.13 ± 0.26 | 97.37 ± 0.35 | 97.98 ± 0.13 | 97.78 ± 0.26 | 98.82 ± 0.10 | 99.33 ± 0.07 | |
| 0.3 | 90.09 ± 0.85 | 95.15 ± 0.61 | 96.69 ± 0.39 | 97.64 ± 0.20 | 96.39 ± 0.46 | 98.65 ± 0.18 | 98.69 ± 0.22 | |
| 0.4 | 79.49 ± 1.71 | 91.01 ± 0.97 | 94.17 ± 0.54 | 96.42 ± 0.58 | 94.98 ± 0.42 | 97.59 ± 0.48 | 98.33 ± 0.56 | |
| 0.5 | 52.39 ± 2.87 | 74.58 ± 2.30 | 90.15 ± 1.60 | 93.69 ± 1.41 | 89.18 ± 1.72 | 96.79 ± 1.53 | 97.73 ± 1.06 | |
| 0.6 | 16.46 ± 1.14 | 34.92 ± 3.69 | 66.92 ± 4.80 | 79.90 ± 4.87 | 53.32 ± 7.81 | 86.58 ± 3.37 | 95.76 ± 1.13 | |
| 0.7 | 6.47 ± 0.71 | 8.12 ± 1.09 | 14.48 ± 2.26 | 42.60 ± 7.60 | 10.85 ± 1.43 | 49.24 ± 7.51 | 79.94 ± 7.87 | |
| 0.8 | 4.81 ± 0.70 | 4.14 ± 0.40 | 2.90 ± 0.34 | 5.39 ± 1.61 | 4.75 ± 0.96 | 35.04 ± 9.04 | 74.52 ± 11.52 | |
| 0.9 | 2.74 ± 0.31 | 3.20 ± 0.28 | 2.83 ± 0.23 | 1.69 ± 0.30 | 2.75 ± 0.45 | 3.58 ± 1.63 | 25.40 ± 7.63 | |
| 1.0 | 2.26 ± 0.23 | 2.57 ± 0.42 | 1.67 ± 0.24 | 1.16 ± 0.08 | 1.87 ± 0.22 | 1.41 ± 0.18 | 3.23 ± 1.51 | |

tion of the dynamics reveals that a region of altruists will tend to expand at the expense of a region of egoists. Altruists have an advantage over egoists in competing for an empty space at the border between regions. The reason for this advantage can be explained as follows: altruists at the border receive help from other altruists in their region, whereas egoists at the border are not supported by other egoists. They might receive help from altruists; however, they face the risk of getting punished.

We conclude that *punishment enhances the evolution of cooperation in more austere environments.*

4.2 Varying the Cost of Punishment

We still focus on our extension of the M_{visc} model. In the first simulations the cost-fine ratio was always 1.0. But what happens if punishment is cheaper or more costly? We performed simulations with varying cost-fine ratios for fixed values $F = 1\%$ and $F = 3\%$. The simulation results are shown in Fig. 4.

We conclude that *increasing the cost of punishment—while keeping CFR constant—enforces the evolution of cooperation.*

4.3 Effects of High-Order Punishment

Further results presented in Table 3 show the number of punishing agents, on the current *Punishment Stage*, among the cooperative population and that for different *Punishment Levels*. On the first punishment level ($\theta = 1$), regarding to the case the CBR is set to the value of 0.5, only 41% of the agents punish the egoists (free riders). Thus about 60% of the agents are first-order free riders, because agents who do not punish are better off than those who punish and pay the cost of punishment.

A comparison of the second punishment level ($\theta = 2$) in the same case only 19% (CBR=0.5, *Punishment Stage 2*) of the population punish the agents who do not punish the egoists. Therefore approximately 80% are second-order free riders. The number of second-order free riders is higher than the number of first-order free riders because the first-order punishment has a greater influence on the egoists than the second-order one. This is obvious, because second-order punishers do not punish egoist directly—but

they punish the agents who do not punish the egoists.

On the third punishment level ($\theta = 3$, *Punishment Stage 3*) 88% are third-order free riders. This supports the assumptions from above. The number of free rides increases with the *Punishment Stage*; therefore high-order punishment is no mechanism to prevent high-order free riders but a mechanism to promote lower-order punishment and thus to promote cooperation. This is also expressed by the results shown in Table 3. The number of agents who punish egoists (first-order punisher) increases with θ . For example, at the first punishment level ($\theta = 1$, CBR=0.5) 41% are first-order punisher, at the second ($\theta = 2$) 55% and at the third ($\theta = 3$) even 82%, which are significant differences.

We conclude that *high-order punishment cannot prevent high-order free-riders.*

4.4 Effects in the Null Model

Hammond and Axelrod demonstrated with their M_{null} model that without local interaction no cooperation can emerge in their agent-based simulation. In a benign environment (in terms of a low cost-benefit ratio, i.e., $C = 1\%$ and $B = 3\%$) the frequency of cooperation is about 4.6%. This result could be expected from further research on the nature of altruistic cooperation.

As mentioned above we also extended the M_{null} model with high-order punishment opportunities. The simulation results indicate that cooperation can emerge more distinctive if punishing is not costly (in terms of a low cost-fine ratio i.e. $P = 0.5\%$ and $F = 1\%$) and help is not costly (in terms of a low cost-benefit ratio i.e. $C = 0.6\%$ and $B = 3\%$). 70% of the agents cooperate in this situation. A graphical presentation of the simulation results (the cost of punishment in this simulation was set to $F=1\%$) is shown in Fig. 5. Our simulation results show that in the M_{null} model higher punishment costs F also increases the cooperation frequency.

We conclude that *without spatial distributed agents cooperation emerges if punishing and help are not costly.*

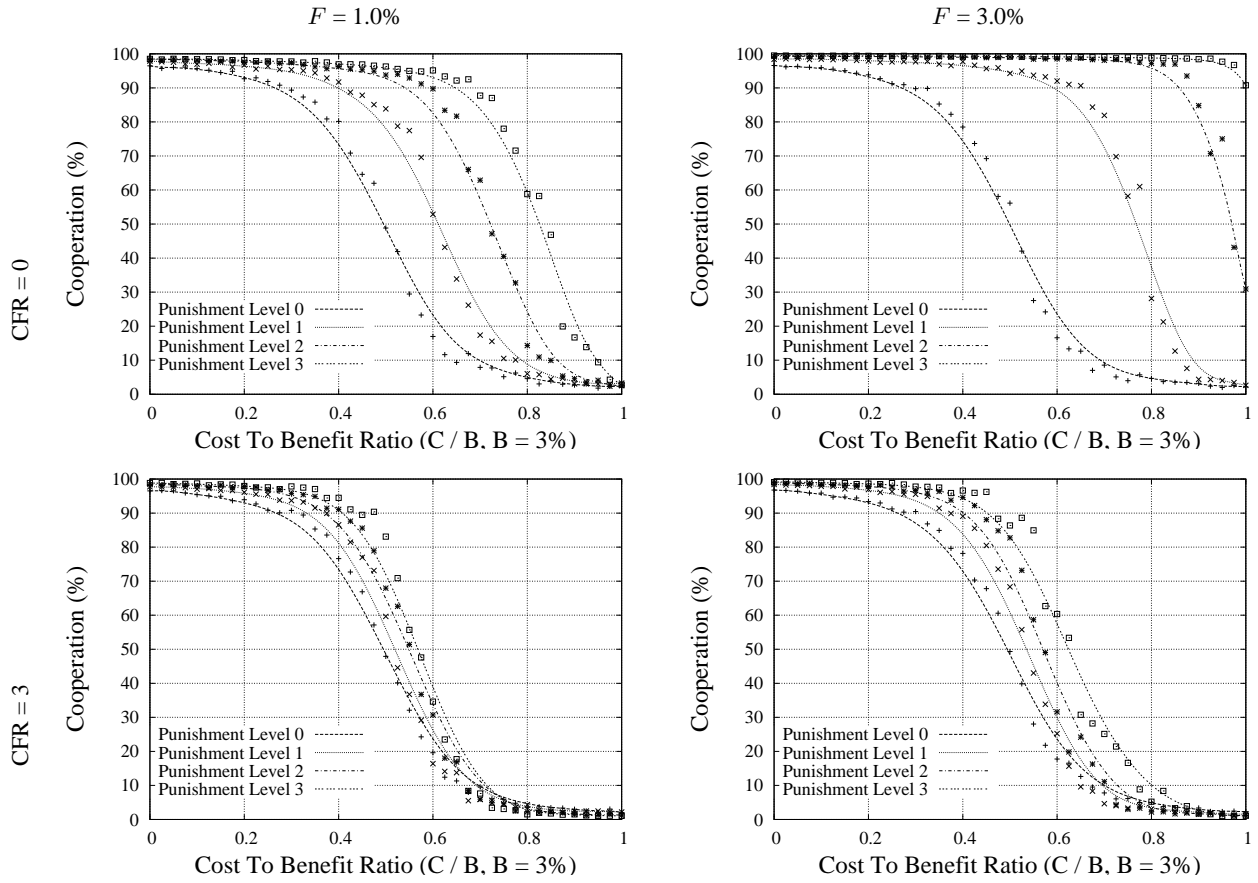


Figure 4: Impact of high-order punishment opportunities on the evolution of cooperation. In a combination of the M_{visc} model (spatially distributed population, limited mobility, local interaction) and the M_{pun} model (opportunity to punish the co-player after being informed about the opponents strategy) higher punishment order still increases the cooperation frequency. This phenomenon could be observed if punishment is not costly (when the CFR is 0.0, i.e., $P = 0.0\%$ and $F = 3.0\%$). P is the cost of punishment for the punisher and F is the cost for being punished. Plus, cross, star, and square symbols depict results from punishment level $\theta = 0, 1, 2,$ and $3,$ respectively. Bezier curves added to illustrate the data trend

Table 3: High-order punishment supports cooperation. It is no mechanism to prevent high-order free riders but a mechanism to promote lower-order punishment (and thus cooperation). Higher punishment levels generate more first-order punishers (Stage 1) within the altruists, when $P = 1.0\%$ and $F = 1.0\%$

| | $\theta = 1$ | | $\theta = 2$ | | $\theta = 3$ | |
|-------|---------------------|---------------------|---------------------|---------------------|---------------|---------------------|
| C/B | Stage 1 (%) | Stage 1 (%) | Stage 2 (%) | Stage 1 (%) | Stage 2 (%) | Stage 3 (%) |
| 0.0 | 46.27 ± 5.36 | 71.72 ± 3.45 | 33.31 ± 4.08 | 84.96 ± 2.47 | 52.46 ± 7.03 | 21.68 ± 6.4 |
| 0.1 | 55.13 ± 4.52 | 60.67 ± 4.38 | 23.82 ± 3.87 | 73.91 ± 8.65 | 42.25 ± 7.36 | 17.1 ± 4.82 |
| 0.2 | 50.03 ± 4.92 | 64.91 ± 5.15 | 28.7 ± 5.44 | 76.78 ± 6.18 | 41.8 ± 7.95 | 15.63 ± 3.62 |
| 0.3 | 44.38 ± 4.25 | 72.04 ± 5.16 | 31.19 ± 5.95 | 81.63 ± 7.27 | 60.93 ± 10.05 | 28.7 ± 7.23 |
| 0.4 | 43.04 ± 4.67 | 72.15 ± 5.64 | 26.26 ± 3.98 | 78.54 ± 6.82 | 46.79 ± 7.2 | 25.08 ± 5.41 |
| 0.5 | 40.83 ± 5.31 | 54.99 ± 6.43 | 18.83 ± 4.06 | 82.31 ± 4.88 | 49.16 ± 5.5 | 11.67 ± 2.12 |
| 0.6 | 31.92 ± 5.34 | 67.33 ± 9.6 | 28.82 ± 5.86 | 78.05 ± 10.87 | 52.88 ± 11.65 | 24.66 ± 7.37 |
| 0.7 | 36.98 ± 8.32 | 64.9 ± 14 | 31.74 ± 10.32 | 83.48 ± 19.96 | 55.88 ± 14.59 | 19.82 ± 5.63 |
| 0.8 | 37.6 ± 8.87 | 38.4 ± 7.45 | 27.43 ± 4.82 | 73.91 ± 30.25 | 66.94 ± 26.93 | 29.32 ± 16.99 |
| 0.9 | 38.82 ± 4.88 | 27.24 ± 5.53 | 29.57 ± 4.3 | 47.96 ± 14.3 | 29.41 ± 6.51 | 27.6 ± 6.2 |
| 1.0 | 45.05 ± 12.92 | 38.12 ± 6.85 | 20.79 ± 3.45 | 36.49 ± 5.35 | 39.19 ± 7.66 | 25 ± 5.52 |

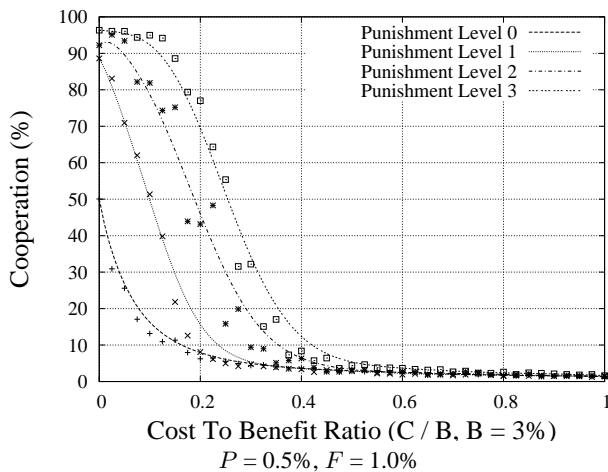


Figure 5: Combining the M_{null} model (no structure in the environment, randomly paired agents) and the M_{ext} model (opportunity to punish co-players) causes to cooperation. Cooperation can emerge more distinctive just when punishing is not costly (in terms of a low cost-fine ratio i.e. $P = 0.5\%$ and $F = 1\%$) and help is not costly (in terms of a low cost-benefit ratio i.e. $C = 0.6\%$ and $B = 3\%$). P is the cost of punishment for the punisher and F is the cost for being punished. Plus, cross, star, and square symbols depict results from punishment level $\theta = 0, 1, 2,$ and $3,$ respectively. Bezier curves were to illustrate the data trend

5. SUMMARY AND OUTLOOK

Based on careful extensions of Axelrod's and Hammond's M_{visc} model, we conclude:

1. High-order punishment enables (and enforces) cooperation in austere environments.
2. Smaller CFR values support the evolution of cooperation.
3. Increasing the punishment F values while keeping the CFR values constant, enforces the evolution of cooperation.
4. High-order punishment cannot prevent high-order free riders. However, it supports the low-order punishments and thus cooperation.
5. Small CFR and CBR values support cooperation in the extended M_{null} model.

The models discussed in this paper are well suited to analyze punishment levels θ from 0 to 3. Higher θ levels reduce the PTR drastically, the corresponding models are not able to analyze punishment. However, basic mechanisms of punishment and cooperation can be studied in these models.

We have discussed various aspects of punishment strategies. Sigmund et al. [11] demonstrated how reward strategies can cause evolution of cooperation. Analyzing the interactions between reward and punishment strategies and their impact on the evolution of cooperation might be of great interest. Again, the problem of decreasing PTR values has to be tackled. Furthermore, it might be interesting to combine the M_{tag} model, which enable cooperation, with punishment strategies.

6. REFERENCES

- [1] B. Beauflis, J.-P. Delhay, and P. Mathieu. Complete classes of strategies for the classical iterated prisoner's dilemma, 2006. http://www.lifl.fr/IPD/references/from_lifl/ep98/html/. Cited January, 25th 2006.
- [2] M. Cinyabuguma, T. Page, and L. Putterman. On perverse and second-order punishment in public goods experiments with decentralized sanctioning. Working Papers 2004-12, Brown University, Department of Economics, 2004. <http://ideas.repec.org/p/bro/econwp/2004-12.html>. Cited December, 5th 2005.
- [3] U. Dieckmann, R. Law, and J. A. J. Metz, editors. *The Geometry of Ecological Interactions: Simplifying Spatial Complexity*. Cambridge Studies in Adaptive Dynamics. Cambridge University Press, Cambridge, UK, 2000.
- [4] M. Doebeli and C. Hauert. Models of cooperation based on the prisoner's dilemma and the snowdrift game. *Ecology Letters*, 8:748–766, 2005.
- [5] E. Fehr and S. Gächter. Cooperation and punishment in public goods experiments. *American Economic Review*, 90(4):980–994, 2000.
- [6] R. A. Hammond and R. Axelrod. Evolution of contingent altruism when cooperation is expensive. <http://www-personal.umich.edu/~axe/>. Cited January, 25th 2006.
- [7] T. Kiyonari and P. Barclay. Second-order punishment and reward in public goods games. In *39th Annual Meeting of the Canadian Economics Association, McMaster University, Hamilton, Canada*, 2005. Poster presentation.
- [8] M. A. Nowak, S. Bonhoeffer, and R. M. May. Spatial games and the maintenance of cooperation. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 91, pages 4877–4881, 1994.
- [9] M. A. Nowak and K. Sigmund. Games on grids. In Dieckmann et al. [3], pages 135–150.
- [10] H. Russell. The free rider problem. The Stanford Encyclopedia of Philosophy, 2003. Cited January, 25th 2006.
- [11] K. Sigmund, C. Hauert, and M. A. Nowak. Reward and punishment. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 98, pages 10757–10762, 2001.
- [12] X. X. et al. The IPD Xxxxx tournament, 2006. <http://www.xxx.xx/>. Cited January, 25th 2006.

C. The Impact of Group-Reputation in a Multi-Agent Environment

The Impact of Group Reputation in Multiagent Environments

Bastian Baranski, Thomas Bartz-Beielstein, Rüdiger Ehlers, Thusinthan Kajendran, Björn Kosslers, Jörn Mehnen, Tomasz Polaszek, Ralf Reimholz, Jens Schmidt, Karlheinz Schmitt, Danny Seis, Rafael Slodzinski, Simon Steeg, Nils Wiemann, Marc Zimmermann

Abstract—This paper presents results from extensive simulation studies on the iterated prisoner’s dilemma. Two models were implemented: a nongroup model to study fundamental principles of cooperation and a model to imitate ethnocentrism. Some extensions of Axelrod’s elementary model implemented individual reputation. To provide a more realistic scenario, we introduced group reputation. We demonstrate that group reputation decreases cooperation with the in-group and increases cooperation with the out-group.

I. INTRODUCTION

The *iterated prisoner’s dilemma* (IPD) is a game for two players, *A* and *B*. It is played for k rounds, where k is unknown to the players. In every round, each player has the option to cooperate or defect with the other player. There is a cost for cooperating but also an advantage if the other player cooperates (Table I). In case of mutual cooperation *A* and *B* receive a *reward* R (3 points). Player *A* receives the *temptation* T (5 points) if *A* defects and *B* cooperates. Player *A* receives zero points, the so-called *sucker’s payoff*, if *A* cooperates while *B* defects. If both players defect, they receive the *punishment* P (1 point). Thus, the entries in the payoff matrix satisfy the following relationship: $T > R > P > S$. Furthermore the following inequality becomes true: $R > (T + S)/2$ (Axelrod, 1984). For very small and known k values, both players will probably defect, because this is best choice in the single-rounded, or non-iterated *prisoner’s dilemma* (PD). This strategy is referred to as *always defect* (ALLD) in the literature. However, playing ALLD results in a lack of the opponent’s confidence and therefore in less points than in case of mutual cooperation. The determination of an optimal strategy, i. e., to maximize the gain at minimum cost, is a difficult task. Its solution depends on several factors. For example, can the players be sure to receive correct information about the opponent or do they have to take into account a certain rate of noise? How often do they play against each other? And can they be sure to play a next round versus the same opponent?

The IPD model can be applied to several real-life situations. Our paper extends models from Axelrod (1997) on agent-based simulations. Axelrod introduced the non-iterated prisoner’s dilemma in an agent-based simulation. He analyzed how agents cooperate or defect against each other on a grid. The agent’s decision whether to cooperate or

Department of Computer Science, Dortmund University, D-44221 Dortmund, Germany (phone: +49-231-755-7705; fax: +49-231-755-7740; The authors of this joint work can be reached via e-mail: `firstname.lastname@udo.edu`).

TABLE I

PAYOFF MATRIX FOR A TWO-PLAYER IPD; S, P, R, T MUST SATISFY $T > R > P > S$ AND $R > (S + T)/2$

| | | |
|-----------|-----------|--------|
| | cooperate | defect |
| cooperate | R | T |
| | R | S |
| defect | S | P |
| | T | P |

to defect influences its *probability to reproduce* (PTR). The decision depends on parameters which are mutated during the game. Two parameters are responsible for the decision, one for the own group and one for every other group, resulting in a friend/foe, or ethnocentric view throughout the game.

Inspired by Axelrod’s work our analysis examines effects and mechanisms influencing the evolution of cooperation. Cooperation is essential in many real-life situations—even among rivals. The cooperation of two rivaling companies is a well-known example: they cooperate to produce new goods, because each company on its own could not have done this. Another example for mutual cooperation is the just-in-time service many suppliers provide, where they deliver their goods in a certain amount to a designated time.

Based thereon, we implemented reputation in Axelrod’s elementary model. Axelrod & Hammond (2003) noted: “To be specific, the model shows that in-group favoritism can overcome egoism and dominate a population even in the absence of [...] reputation [...]” This statement was subject of our investigations. Reputation plays an important role in today’s life. In our society a good reputation can often change people’s behavior. An example from real life is the commercial online-marketplace eBay where each user gets an own account. This account enables him or her to buy and sell all kinds of goods. Each member also has a feedback score, which tells other users how reliable this member was up to now. If this score is very low other members would possibly be more cautiously dealing with this member. Marler & Evans (1996) and Zahavi & Zahavi (1997) described the influence of renown on human (and also animal) behavior. By increasing one’s own behavior in the sense of being more altruistic one can increase the chance of being the subject of further altruistic acts by other individuals.

Experiences from the Dortmund IPD tournament were integrated into our studies. The Dortmund tournament was organized and analyzed by the authors in 2005. More than 50

different strategies took part in this contest, more information can be accessed under the given url. By memorizing the decisions in the past we were able to mirror the circumstances in the conventional IPD for the agents in Axelrod’s simulation, where the agents were confronted with the circumstances of the PD. The tournament and the simulation software used in this paper can be used for further experiments or to reproduce results presented in this study. The software from this study is freely available (PG 474, 2005).

We propose a model to examine the working principles and the impact of reputation. Therefore agents were equipped with the knowledge from preceding rounds. Results from simulations based on these modified setups are compared to results from Axelrod’s original experiment. Similar to approaches described in Yao & Darwen (2000), Chess (1988), and Fogel (1993), co-evolutionary learning was used in the IPD to determine proper settings for different parameters. We investigated the evolution of cooperation and the effects of reputation thereon.

Axelrod’s model was extended in two different ways. In the first model M_1 , we removed the affiliations to the different groups from the agents; i.e. all agents belonged to the same group, thus played for themselves alone because there were no other groups. Furthermore the agents had the ability to remember the behavior of the other agents. This memory included information on how each other agent had played so far, not only against a specific agent but against all agents. Memory enabled us to give each agent a reputation, based solely on his behavior during the current game. This reputation likewise influenced the behavior of all other agents towards this agent, so a good reputation of an agent, i.e., one who had often cooperated, resulted in a more positive behavior of the others towards himself and vice versa.

Our second model M_2 , used an enhanced representation of reputation. Again, agents belonged to different groups. Moreover the agents had the ability to differentiate all existing groups on the grid. They had the opportunity to work together with specific other groups. Finally we combined this differentiating view with our concept of reputation to observe the resulting effects.

Our reputation model extends other implementations. Yao & Darwen (2000) used a tag to describe how an agent had behaved so far, but this tag was initialized with 0 and could take on during the game only the values “-1” and “1”, while our reputation score could take on every value ranging from 0 to 1 in 0.1 steps. Nowak & Sigmund (1998) used an image score, which was initialized with 0 and increased or decreased by one depending on the decisions the agent has made so far. This is in contrast to our model that stores the percentage of cooperation for each agent. Schenk (1995) used another reputation scheme: he equipped each agent with memory to collect information on how this agent was treated so far. However, the agent’s behavior, and not its treatment, is memorized in our model to determine its current reputation. We equipped each agent with a memory of how all other agents played so far.

This paper is organized as follows: The reputation models are introduced in Sect. II. Section III describes results from the experiments, Sect. IV presents an analysis. Before an outlook is given in Sect. VI, conclusions are drawn in Sect. V.

II. DEFINITION

Axelrod (1997) tried to discover important factors influencing the evolution of cooperation and ethnocentrism. In his model, agents were placed on a grid and played against each other. Successful agents receive a higher rate to bring forth offspring.

Our goal was to define a model that enables agents to build up a kind of reputation, i.e., a way in which they knew how well or badly their opponent had behaved so far. Knowledge gained by the agents so far was used for this purpose. Each agent has to play several rounds of the PD against several opponents. Our model of reputation used the knowledge gained during these games to create a kind of renown for each agent or, as explained later on, for each group. Furthermore each agent used a parameter which determines the next move. This decision depends on the opponent’s reputation. Two variants of this reputation model were implemented. Both variants extend the basic environment from Axelrod (1997) which is explained in Sect. II-A. Model M_1 is a less complicated extension, in which every agent is reliant on himself only as there are no longer any groups. Model M_2 implements different groups and uses a modified reputation scheme.

A. Basic environment

The basic environment bases upon the original experiments (Axelrod, 1997). We use a grid \mathcal{G} on which agents from different groups are randomly placed. This grid is a swap around square with a certain side length, overlapping at the borders in a way that each cell on the left border is a neighbor to the corresponding cell on the right border. In the same way upper and bottom cells are treated. The *neighborhood* of an agent at position p_i is defined as the set of grid points that can be reached from p_i with one vertical or horizontal step.

Each cycle in the simulation consists of four phases: immigration, interaction, reproduction, and death. In the immigration phase an agent with stochastic characteristics is placed randomly on a free cell and every agent receives a certain *probability to reproduce* (PTR). Neighbors (horizontal or vertical) play one round of the Prisoner’s Dilemma against each other. Their decision to cooperate or to defect depends on the opponents’ reputations.

Let γ denote a constant which models the *costs of cooperation*. Each agent has to pay a certain amount γ from his PTR when cooperating, while paying none when defecting: Consider two agents, a_i and a_j . The PTR value of an agent a_i who cooperates is reduced by γ , if a_j defects. If the opponents a_j cooperates, a_i receives a bonus δ increasing his probability to reproduce. So if both opponents cooperate, each receive a bonus that increases their PTR values.

TABLE II

PARAMETER SETTINGS IN BASIC ENVIRONMENT. $\mathcal{G}_{\text{SIZE}}$ = SIDE LENGTH OF THE GRID, γ = COST TO COOPERATE, PTR = PROBABILITY TO REPRODUCE, δ = AMOUNT OF PTR GAINED WHEN OPPONENT COOPERATED, p_{MUT} = MUTATION RATE, p_{DIE} = RISK TO DIE. FROM *Evolution of Ethnocentric Behavior* AXELROD & HAMMOND (2003)

| $\mathcal{G}_{\text{size}}$ | γ | PTR | δ | p_{mut} | p_{die} |
|-----------------------------|----------|------|----------|------------------|------------------|
| 50 | 1.0 | 12.0 | 3.0 | 0.5 | 10.0 |

In the reproduction phase offspring are placed randomly on a directly neighboring cell. Parameters are mutated with a so-called *mutation rate* p_{mut} . The parameters which are exposed to a possible mutation are the group of the agent as well as the parameters concerning the reputation.

In the death phase every agent which was at the beginning of this turn on the board and each immigrated agent (except the offspring) have a fixed *risk of dying* p_{die} . After the dying phase the PTR of each agent is reset to its initial value. To enable a comparison with other models, the model parameters have been chosen as proposed by Axelrod & Hammond (2003)). The experimental setup is summarized in Table II.

B. Nongroup model M_1

Reputation can be interpreted as “the overall quality as seen or judged by people in general.” Thus, reputation requires memory. To introduce reputation, agents were equipped with memory. We define reputation according to the number of cooperations this agent has chosen in the PD game so far. In this first model every agent has an individual reputation, meaning every agent’s decisions influences his reputation directly. For this purpose different groups were eliminated, so each agent played only for himself, not his group. The following two parameters were introduced in the M_1 model: The *reputation* ρ_{indi} is defined as the relation between the number of cooperations and the number of total moves m an agent a had so far

$$\rho_{\text{indi}}(a) = \text{round}(c(a)/m(a))/10. \quad (1)$$

ρ_{indi} ranges from 0.0 to 1.0 in 0.1 steps, where 1.0 means total cooperation, or, said in another way, the agent had cooperated in nearly every move up to now, while 0.0 means this agent had never cooperated during the whole game so far.

The second parameter, the so-called *reputation threshold* τ_{indi} , determines the lowest level of reputation which is necessary to cooperate with another agent. Consider agent a_i with $\rho_{\text{indi}} = 0.3$ and agent a_j with $\tau_{\text{indi}} = 0.6$: a_j does not cooperate with a_i , because $\rho_{\text{indi}}(a_i) < \tau_{\text{indi}}(a_j)$. Agent a_j did not cooperate with a_i because of a_i ’s bad reputation.

Note, τ_{indi} is only altered via mutation. These two parameters were randomly initialized for every agent and they also were subject to a possible mutation while passing on to a child. We tried to verify our results by using different initializations, like high expectations ($\tau_{\text{indi}} = 1$) and a low

own reputation ($\rho_{\text{indi}} = 0$) and vice versa, the results will be presented in the next chapter.

C. Multi-group model M_2

The second model analyzes different groups and the ability to differentiate between them. The reputation from model M_1 implemented for each group, meaning that the average number of cooperations of every agent belonging to the group was considered. This parameter is referred to as *groupReputation* ρ_{group} . In this way the individual decision of one agent only contributed partly to the reputation of the whole group. The parameter ρ_{indi} is kept, only the determination of the reputation differed from the first model: In the first model each agent’s reputation was calculated on the basis of his behavior only, while in the second model the reputation (of the whole group) was averaged over each reputation of all agents of this group. Let n_j denote the number of agents in the j th group (with $j \in [1; m]$) ρ_{group} can be calculated as

$$\rho_{\text{group}}(j) = \frac{1}{n_j} \sum_{i=1}^{n_j} \rho_{i,j}. \quad (2)$$

The parameter τ_{indi} was extended, so that there was an own τ_{indi} for every agent of every occurring group: $\tau_{i,j} \in [0.0; 1.0]$.

Ethnocentrism was implemented in model M_2 as follows: An *ethnocentric agent* a_{ethno} i is an agent that defected at least once against an agent b of another group j (against an out-group, after Axelrod). At the same time a ’s $\tau_{\text{indi}}(a, i)$ against this group has to be less than 0.6, i. e.

$$a_{\text{ethno}}(i) = \begin{cases} 1 & \tau_{\text{indi}}(a, i) \leq 0.6 \text{ and } d \geq 1 \\ 0 & \text{otherwise,} \end{cases}$$

where d denotes the number of defections against outgroups. While playing against other agents from the own group (in-group), agents have to fulfil the following criterion: His τ_{indi} concerning his own group must have been lower or equal to the τ_{group} of his group. This means, the agent cooperated with his own group every time he played against another agent of his own group.

Obviously, M_1 does not model ethnocentrism because it implements no groups. Each simulation contained 2000 generations and we ran each simulation 10 times, using not only the average results over the last 100 generations for our analysis but also the average over all 2000 generations. In addition we also tested model M_2 regarding more groups (6 and 8 groups). Furthermore we examined which effects would take place when the costs (in PTR) for cooperating are lower (0) and higher (2 and 3) in order to validate our results. To enable a comparison of our results to the results from Axelrod & Hammond (2003), we produced the same tables.

III. RESULTS

A. Individual reputation in a non-group-environment

In an environment in which all agents belong to the same group, meaning there is no distinction between groups at all,

TABLE III

FIVE DIFFERENT CONFIGURATIONS FOR INITIAL ρ_{INDI} , ρ_{GROUP} AND τ_{INDI}

| ρ_{indi} | ρ_{group} | τ_{indi} | configuration |
|----------------------|-----------------------|----------------------|---------------|
| 0 | | 0 | κ_0 |
| 0 | | 1 | κ_1 |
| 1 | | 0 | κ_2 |
| 1 | | 1 | κ_3 |
| random $\in [0; 1]$ | | random $\in [0; 1]$ | κ_4 |

every agent possesses an own reputation ρ_{indi} and a single reputation threshold τ_{indi} against all other agents. ρ_{indi} is built up during the past rounds and its initialization was varied in our experiments. The τ_{indi} , which was also initialized with different values, stayed fix during a run and mutated only if inherited to offsprings. The following parameters were modified: ρ_{indi} , τ_{indi} , and the costs to cooperate γ , which are subtracted from the PTR of the cooperating agent. The different parameter settings for ρ_{indi} and τ_{indi} are shown in Table III. In addition to the values in this table the costs to cooperate γ were also modified. According to the original experiments in Axelrod & Hammond (2003), γ was set to 0, 1, 2 and 3. Lower values, e.g. 0 or 1, represent a friendly environment, higher values, e.g. 2 or 3, represent more hostile environments. Regarding the amount of help gained, which means a value of 3 to be added to the PTR, when the opponent had cooperated, there would be little or no benefit in cooperating in a hostile environment. Each of the 20 different settings were repeated 10 times. The mean value from these runs were taken to compare our results with the ones presented in Axelrod & Hammond (2003).

We compare the results of selected experiments of our own with the according results from the experiments Axelrod and Hammond conducted. Interesting values were printed in boldface. Table IV shows different values of the percentage of behavior: Cooperation with the same group depends on the five different configurations for initial ρ_{indi} and τ_{indi} introduced in Table III. C_{same} represents the percentage over all 2000 generations and $C_{\text{same}}^{\text{last } 100}$ the percentage only over the last 100 generations. For both values the differences ΔC_{same} and $\Delta C_{\text{same}}^{\text{last } 100}$ to the results from Axelrod & Hammond (2003) are shown. In Table IV results from the experiment with $\gamma = 1$ are compared to experiment no. 117 from Axelrod & Hammond (2003). In Table V results from the experiment with $\gamma = 2$ is compared to experiment no. 119 of Axelrod & Hammond (2003).

The comparison in Table IV demonstrates how the ratio of cooperation within the same group (in this model there was only one group) varied only insignificantly from the results in Axelrod & Hammond (2003), when unfavorable initial values for ρ_{indi} and τ_{indi} are chosen. Beneficial starting values however boosted the increase of cooperation in a significant way. Even a random initialization of ρ_{indi} and τ_{indi} shows a significant increase in the ratio between the number of cooperations and the number of overall interactions in the whole game.

Table V shows the positive effect of reputation on coop-

TABLE IV

COMPARING THE PERCENTAGE OF BEHAVIOR THAT IS COOPERATION WITH SAME TYPE TO RESULTS FROM AXELROD & HAMMOND (2003). $G = 50$, $\gamma = 1$, PTR = 12.0, $\delta = 3.0$, $p_{\text{MUT}} = 0.5$, $p_{\text{DIE}} = 10.0$, 1 GROUP, SEE TABLE II FOR EXPLANATION. C_{SAME} IS THE ABSOLUTE VALUE OF PERCENTAGE OF BEHAVIOR THAT IS COOPERATION WITH SAME TYPE OVER WHOLE GAME, $C_{\text{SAME}}^{\text{LAST } 100}$ IS THE SAME LIKE C_{SAME} BUT OVER THE LAST 100 ROUNDS ONLY, ΔC_{SAME} AND $\Delta C_{\text{SAME}}^{\text{LAST } 100}$ ARE THE RELATIVE DIFFERENCES TO THE VALUES FROM AXELROD & HAMMOND (2003)

| configuration | κ_0 | κ_1 | κ_2 | κ_3 | κ_4 |
|---|------------|---------------|---------------|--------------|------------|
| C_{same} | 97.41 | 75.63 | 97.65 | 75.44 | 85.34 |
| ΔC_{same} | +20.81 | -0.97 | +21.05 | -1.16 | +8.74 |
| $C_{\text{same}}^{\text{last } 100}$ | 97.39 | 93.07 | 97.49 | 94 | 94.25 |
| $\Delta C_{\text{same}}^{\text{last } 100}$ | +22.09 | +17.77 | +22.19 | +18.7 | +18.95 |

TABLE V

COMPARING THE PERCENTAGE OF BEHAVIOR THAT IS COOPERATION WITH SAME TYPE WITH RESULTS FROM AXELROD & HAMMOND (2003), SAME AS IN TABLE IV BUT WITH COST DOUBLED TO $\gamma = 2$

| configuration | κ_0 | κ_1 | κ_2 | κ_3 | κ_4 |
|---|------------|---------------|---------------|---------------|------------|
| C_{same} | 94.64 | 37.08 | 95.34 | 39.49 | 67.63 |
| ΔC_{same} | +71.24 | +13.68 | +71.94 | +16.09 | +44.23 |
| $C_{\text{same}}^{\text{last } 100}$ | 94.93 | 72.37 | 94.48 | 74.34 | 85.97 |
| $\Delta C_{\text{same}}^{\text{last } 100}$ | +80.93 | +58.37 | +80.48 | +60.34 | +71.97 |

eration. In this configuration the costs to cooperate γ were doubled, thus the environment was modeled in a more hostile way. Despite this unfavorable starting-point there still is a larger ratio of cooperation than in Axelrod & Hammond (2003), even in runs, when the initial configuration of ρ_{indi} and τ_{indi} were adversarial for cooperation.

B. Group-reputation in a group-environment

In this environment agents also possessed a kind of reputation. Each group had a reputation value ρ_{group} , which was determined by the decisions of each group member. Furthermore each agent possessed a reputation threshold τ_{group} for each group on the grid. The starting values for ρ_{group} , γ , and τ_{group} for each other group (or color) were again varied, see Table III. In addition the number of groups was varied, too. We conducted experiments with four, six and eight groups. Our results were compared to the results from Axelrod & Hammond (2003).

In Table VI the ratio C_{same} (and $C_{\text{same}}^{\text{last } 100}$ respectively for the last 100 generations) between the number of cooperations and the number of total interactions between agents of the same group again was compared to the values found in Axelrod & Hammond (2003). In this case results were always lower than the results of the corresponding experiments. This means, that the number of cooperations with agents of the same group decreased in our experiments. The other values depicted in Table VI are percentage of behavior that is defection with different type D_{diff} and $D_{\text{diff}}^{\text{last } 100}$, which is the same but for the last 100 generations only. Compared to results from experiment no. 104 in Axelrod & Hammond (2003) it is obvious that the amount of defections with agents

TABLE VI

COMPARING THE PERCENTAGE OF BEHAVIOR THAT IS COOPERATION WITH SAME TYPE AND OF BEHAVIOR THAT IS DEFECTION WITH DIFFERENT TYPE TO RESULTS FROM AXELROD & HAMMOND (2003). THE SAME SETTINGS AS IN TABLE IV WERE USED, BUT FOUR GROUPS HAVE BEEN SIMULATED. C_{SAME} IS THE ABSOLUTE VALUE OF PERCENTAGE OF BEHAVIOR THAT IS COOPERATION WITH SAME TYPE OVER WHOLE GAME, $C_{\text{SAME}}^{\text{LAST } 100}$ IS SIMILAR TO C_{SAME} BUT OVER THE LAST 100 ROUNDS ONLY, D_{DIFF} IS THE ABSOLUTE VALUE OF PERCENTAGE OF BEHAVIOR THAT IS DEFECTION WITH DIFFERENT TYPE OVER WHOLE GAME, $D_{\text{DIFF}}^{\text{LAST } 100}$ IS THE SAME BUT OVER THE LAST 100 ROUNDS ONLY, ΔC_{SAME} AND $\Delta C_{\text{SAME}}^{\text{LAST } 100}$ ARE THE DIFFERENCES RELATIVE TO THE VALUES OF AXELROD & HAMMOND (2003), ΔD_{DIFF} AND $\Delta D_{\text{DIFF}}^{\text{LAST } 100}$, RESPECTIVELY

| configuration | κ_0 | κ_1 | κ_2 | κ_3 | κ_4 |
|---|------------|---------------|---------------|------------|------------|
| C_{same} | 70.91 | 49.99 | 71.49 | 62.77 | 73.96 |
| ΔC_{same} | -17.39 | -38.31 | -16.81 | -25.53 | -14.34 |
| $C_{\text{same}}^{\text{last } 100}$ | 70.35 | 77.76 | 70.84 | 68.05 | 77.06 |
| $\Delta C_{\text{same}}^{\text{last } 100}$ | -19.45 | -12.04 | -18.96 | -21.75 | -12.74 |
| D_{diff} | 3.61 | 11.50 | 0.12 | 6.68 | 6.69 |
| ΔD_{diff} | -72.39 | -64.50 | -75.88 | -69.32 | -69.31 |
| $D_{\text{diff}}^{\text{last } 100}$ | 6.14 | 7.57 | 0.50 | 9.49 | 8.93 |
| $\Delta D_{\text{diff}}^{\text{last } 100}$ | -77.59 | -76.15 | -82.23 | -74.24 | -74.80 |

TABLE VII

SAME AS IN TABLE VI BUT WITH COSTS DOUBLED TO $\gamma = 2$

| configuration | κ_0 | κ_1 | κ_2 | κ_3 | κ_4 |
|---|------------|---------------|---------------|------------|------------|
| C_{same} | 70.82 | 2.66 | 73.29 | 33.20 | 58.16 |
| ΔC_{same} | +1.32 | -66.84 | +3.79 | -36.3 | -11.34 |
| $C_{\text{same}}^{\text{last } 100}$ | 66.42 | 4.58 | 73.08 | 19.35 | 55.11 |
| $\Delta C_{\text{same}}^{\text{last } 100}$ | -2.98 | -64.82 | +3.68 | -50.05 | -14.29 |
| D_{diff} | 5.76 | 20.86 | 0.31 | 13.17 | 12.44 |
| ΔD_{diff} | -78.54 | -63.44 | -83.99 | -71.13 | -71.86 |
| $D_{\text{diff}}^{\text{last } 100}$ | 9.75 | 21.59 | 0.84 | 19.51 | 14.01 |
| $\Delta D_{\text{diff}}^{\text{last } 100}$ | -79.01 | -67.17 | -87.92 | -69.25 | -74.75 |

of a different color decreased massively. Thus there is less defection on the grid.

In Table VII we compared the results from experiment no. 109 in Axelrod & Hammond (2003) to results from our simulations. The γ value was doubled again. The influence of reputation was high enough to reach a slightly better result than the corresponding experiment in Axelrod & Hammond (2003), even if only in configurations κ_0 and κ_2 . These configurations are beneficial for cooperation. The D_{diff} none the less is much lower than the one in Axelrod & Hammond (2003), showing that even under harsh conditions (i. e. $\gamma = 2$) there is less defection due to the factor of cooperation.

Increasing the number of groups has a negative effect on the behavior towards the own group. In Table VIII we compare the model of reputation with eight groups to the corresponding experiment in Axelrod & Hammond (2003). Independently of the quality of the initial configurations of ρ_{group} and τ_{group} the results are inferior regarding the amount of cooperation with the own group. A comparison of different D_{diff} values demonstrates the impact of reputation on defection.

TABLE VIII

SAME AS IN TABLE VI BUT WITH NUMBER OF GROUPS DOUBLED TO 8

| configuration | κ_0 | κ_1 | κ_2 | κ_3 | κ_4 |
|---|------------|---------------|---------------|---------------|------------|
| C_{same} | 66.44 | 55.56 | 66.68 | 60.05 | 69.14 |
| ΔC_{same} | -23.76 | -34.64 | -23.52 | -30.15 | -21.06 |
| $C_{\text{same}}^{\text{last } 100}$ | 65.71 | 77.85 | 64.73 | 74.96 | 69.29 |
| $\Delta C_{\text{same}}^{\text{last } 100}$ | -24.79 | -12.65 | -25.77 | -15.54 | -21.21 |
| D_{diff} | 3.88 | 13.07 | 0.33 | 10.76 | 8.66 |
| ΔD_{diff} | -73.41 | -64.22 | -76.95 | -66.53 | -68.63 |
| $D_{\text{diff}}^{\text{last } 100}$ | 5.29 | 8.87 | 0.87 | 9.01 | 8.86 |
| $\Delta D_{\text{diff}}^{\text{last } 100}$ | -79.95 | -76.36 | -84.37 | -76.22 | -76.37 |

C. Which factors influence the impact of reputation the most

Our experiments showed, that the above defined influence of reputation towards evolution of cooperation are steady, with external factors like configurations κ , cost γ , or number of groups m only changing the amplitude of differences but not the trend of decreasing ethnocentrism and thus increasing cooperation between the agents. κ_2 is a noteworthy exception concerning the variable γ . As can be seen in Fig. 1, not even γ , which defines the harshness of the environment, and therefore is an indicator for the amount of cooperative behavior, influences the steady high value for κ_2 . Figure 1 shows the percentage of cooperation with in-group C_{same} , it should be annotated, that this result is coherent with all other statistically surveyed values, i. e. D_{diff} , $C_{\text{same}}^{\text{last } 100}$, and $D_{\text{diff}}^{\text{last } 100}$, respectively.

IV. ANALYSIS

A. Analysing model M_1 , one-group reputation

First of all we will divide the configurations from Table III into two groups: the more positive starting points κ_0 and κ_2 regarding the initial values of ρ_{indi} and of τ_{indi} will henceforth be referred to as χ_{pos} . The worse starting points κ_1 and κ_3

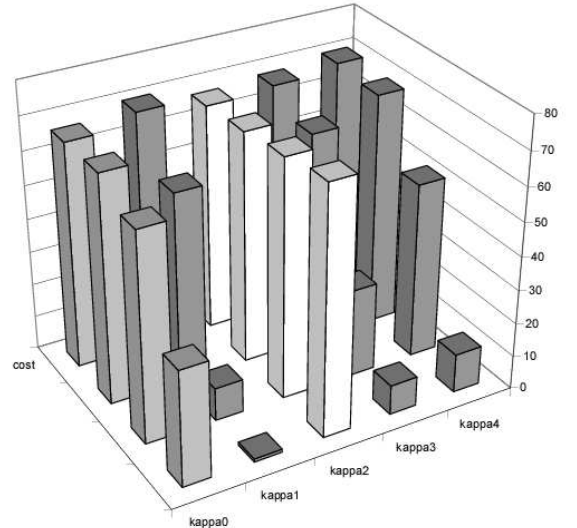


Fig. 1. Percentage of cooperation with in-group C_{same} , cost γ is lowest in rear row increasing to the front row, see Sect. III-C

will be referred to as χ_{neg} . This arrangement will be held during the analysis of both models.

One might conclude that under χ_{neg} conditions reputation reduces the number of cooperations compared to Axelrod (where ΔC_{same} is negative), see Table IV. However, regarding only the values taken over the last 100 rounds clearly shows that our model of reputation requires some time to improve the amount of cooperation. The $C_{\text{same}}^{\text{last } 100}$ value again is higher than in Axelrod’s model; meaning that in our model of reputation under the bad starting conditions χ_{neg} there is in the beginning less cooperation compared to Axelrod’s simulation—but the increase in cooperation is higher, resulting in a greater amount of cooperation towards the end of the simulation.

Regarding χ_{pos} displays the effect of reputation compared to the results from Axelrod. Here also, especially towards the end of our simulation, the increase in cooperation is obvious and noticeably higher than in Axelrod’s simulation. Considering this it is very likely that in longer simulations the amount of cooperation would be higher due to reputation and it’s long-term effect.

Table V displays the same settings and comparisons as shown in Table IV with doubled costs γ . Here the effects of reputation are even more obvious. While higher costs represent a more hostile environment in itself, even under bad starting conditions χ_{neg} is a higher amount of cooperation compared to Axelrod’s results. Towards the end of the simulation this increase is even higher, thus showing that our model of reputation to a certain extent can compensate several effects which would otherwise decrease the amount of cooperation. This is shown also in a comparison between both Table IV and Table V. While the values of C_{same} under χ_{neg} in Table V each is approximately one half of the corresponding values in Table IV, at the end of the simulation ($C_{\text{same}}^{\text{last } 100}$) the difference is considerably smaller, proving that there was a compensation during the simulation.

Summarizing, we can state that on the one hand reputation can compensate—at least to a certain extent—the negative factors in our experiment. On the other hand reputation has an overall positive effect on the amount and on the increase of cooperation.

B. Analyzing model M_2 , multi-group reputation

While comparing Table VI and VIII one striking fact is obvious: the number of groups does not influence the level of cooperation. The values in Table VIII do not vary much compared to the corresponding values of Table VI. Regarding the level of defection with other groups in Table VI (D_{diff}) displays a high amount of cooperation with the out-group’s while C_{same} displays a somewhat smaller amount of cooperation with the in-group. Logically consistent does the number of out-group’s have no or only a small effect on the amount of cooperation overall and on the ratio of cooperation. The reasons for the relatively high amount of defection with the in-group under these conditions will be subject of further simulation studies.

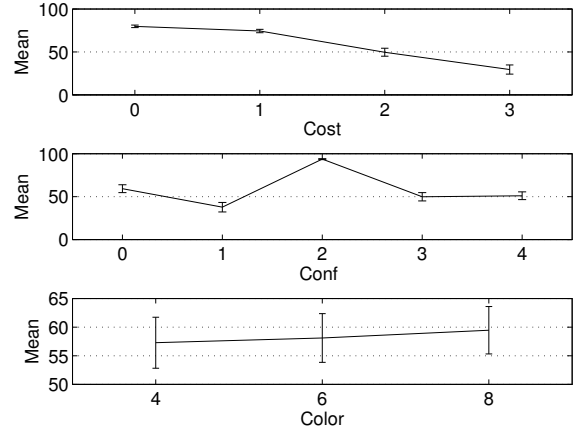


Fig. 2. Design plots (with 95% confidence intervals) illustrating the effects of the factors cost, mode, and color on the cooperation level Y . Design plots cannot illustrate interactions between factors, they show only their main effects. Configuration has the largest effect, whereas the choice of the color is negligible

In Table VII the amount of cooperation under κ_2 is highest (C_{same} high, D_{diff} small) and only marginally smaller than the corresponding values in Table VI. Even though the costs γ are doubled (Table VII) the positive effect of κ_2 is enough to compensate this negative influence.

Simulating κ_1 leads to a significantly lower amount of cooperation than in every other configuration, especially with the in-group. This effect is caused by the starting values of ρ_{group} and τ_{group} . Starting with $\rho_{\text{group}} = 0$ and $\tau_{\text{group}} = 1$ means that all agents at first did not cooperate with one another. “They believed the worst of others and did nothing to improve their own reputation (or the reputation of their group)” would be a good description of their behavior.

C. Analysis using regression trees

To quantify the level of cooperation, we introduced the following measure: The *cooperation level* Y is the percentage of agents with $\rho_{\text{indi}} \geq 0.6$.

$$Y = \frac{1}{mT} \sum_{t=1}^T \left\{ \sum_{j=1}^m \left(\sum_{i=1}^{n_t} x_{ij} \right) \frac{1}{n_t} \right\}, \quad (3)$$

where

$$x_{i,j} = \begin{cases} 1 & \rho_{i,j} \geq 0,6 \\ 0 & \rho_{i,j} < 0,6. \end{cases} \quad (4)$$

Tools from explorative data analysis and computational statistic were used to screen out important factors of the simulation models. Design plots vary one factor while averaging the response over the other factors. Model M_2 has three important factors: cost, configuration, and color. The design plots from Fig. 2 depict their influence on Y . Therneau & Atkinson (1997) describe regression trees as flexible non-parametric tools for screening variables. The same data as in Fig. 2 have been used to generate the tree in Fig. 3.

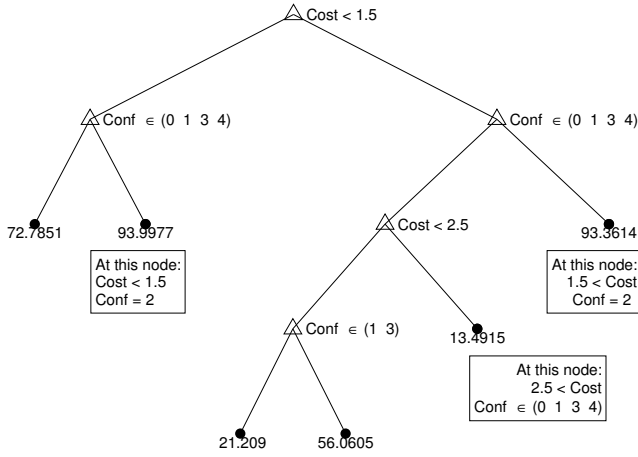


Fig. 3. Regression tree to determine significant parameter settings. This tree complements results from the design plot (Fig. 2). The factor at the root node (Cost) has the greatest effect on the response Y . However, high cooperation levels can be obtained if κ_2 is chosen, independently of the costs. This can be seen from the nodes of the third level in the tree

When looking closer at the regression trees, one can see that configuration κ_2 causes a high Y -value, not depending on the parameter γ . Even in a hostile environment with $\gamma = 3$, Y stayed over 90%. A possible explanation is rooted in the start values of ρ_{indi} and τ_{indi} . New agents get equipped with a ρ_{indi} of 1 and other agents will cooperate when interacting with a τ_{indi} of 0. It seems probable that most agents and their children, who inherit reputation-level and cooperation-threshold do not live long enough to gain a bad reputation. So even in the configuration κ_2 , where an agent would gain a maximum of 3 points to his PTR while paying 3 points every time he cooperates, agents tend to cooperate with their neighbors on the grid, caused by their low τ_{indi} and the high ρ_{indi} of any other agent.

For the configurations κ_1 and κ_3 , the value of γ is decisive. The higher γ , the lower the value of Y gets, which was quite predictable. In both configurations agents got a τ_{indi} of 1 when placed on the grid and it obviously takes some time to lower the threshold. The percentage of cooperations in κ_1 is lower than that in κ_3 , which is caused by the value of ρ_{indi} (0 in κ_1 , 1 in κ_3).

In experiments with configuration κ_0 , which works with start values of $\rho_{\text{indi}} = \tau_{\text{indi}} = 0$ and thus is comparable to κ_3 with $\rho_{\text{indi}} = \tau_{\text{indi}} = 1$, the rate of cooperative agents receives a significantly higher value even for high γ values, being not independent from the number of groups on the grid (experiments using κ_0 with $\gamma = 3$ produced results of 10%, 17% and 26% with 4, 6 and 8 groups, while κ_3 results were all about 18%-19% for experiments with a different number of groups). This is even more surprising if one looks at the conditions which lead to the value of Y : in κ_0 , new agents do not get summed up into Y because of their low default value of ρ_{indi} . It may take some generations for agents to get a ρ_{indi} over 0,6 (which is essential for a high group reputation), but it's easier for agents in κ_0 to get into cooperative interaction,

because it's easier for an agent to gain ρ_{indi} than it's to lower the τ_{indi} -Parameter, which is just influenced by mutation, not by interactive behavior.

Further experiments will be performed to gain knowledge about possible reasons for the high values in κ_0 and interaction with the number of groups.

V. CONCLUSIONS

The conclusion is separated into different thesis and ordered according to the underlying model.

A. Individual reputation in a multi-agent non-group environment

1) *Reputation encourages cooperation:* In M_1 , which uses only one group, the increase in the percentage of cooperation was significant. While the configurations χ_{neg} for ρ_{indi} and τ_{indi} come off even with slightly less cooperation over the whole 2000 rounds, the cooperative behavior differed only sparsely in the final 100 rounds.

2) *Impact of reputation on cooperation is more obvious when costs increase:* With doubled costs for M_1 the impact of reputation on cooperation is higher. Even the χ_{neg} achieves higher percentages over the whole 2000 rounds. Results from the last 100 rounds again are close to the results from χ_{pos} .

3) *Unfavorable initial values slowly develop to higher cooperation:* The initial values for ρ_{indi} and τ_{indi} are decisive for the development of cooperation. Even though in each configuration a steady state will be reached at time, the amount of rounds to be played by agents with unfavorable initial settings is much higher than with favorable or random settings.

B. Group-reputation in a multi-agent environment

1) *Group-Reputation encourages cooperation with out-group while discouraging cooperation with in-group :* The presence of group-reputation increases cooperative behavior vis-à-vis agents of different colors. Even increasing costs for the altruistic act of cooperating cannot delay this effect, nota bene a really harsh environment (e. g., $\gamma \geq 3$) still reduces cooperation, but for the in-group respectively. However, group-reputation seems to decrease cooperation with the in-group, while the number of groups has no significant influence for this value, the presence of groups principally reduces cooperation with the in-group.

2) *Number of groups does not affect cooperation with in-group :* While there is an effect for the defection towards the out-group by the number of groups, there is no such effect on the cooperation with the in-group.

3) *Believe in good nature (κ_2) is not affected by hostile environment, but believe in bad nature decreases cooperative behavior:* Configuration κ_2 , using the values of $\rho_{\text{indi}} = 1$ and $\tau_{\text{indi}} = 0$, is not affected by a high γ . Even with $\gamma = 3$ the percentage of cooperative behavior Y stays beyond 90%. Agents believing in the good nature of others tend to cooperate, even if their PTR cannot be increased by cooperating with others. On the other hand, in experiments using the configurations κ_1 and κ_3 (where new agents starts with a

τ_{indi} of 1, which can be interpreted as a certain skepticism) the cooperative behavior decreases with an increasing γ . The τ_{indi} is only affected by mutation and it obviously takes too long for agents and the children to change their opinion.

4) *The number of groups affects cooperative behavior under certain circumstances* (κ_0): In a hostile environment the Y -value increases with the number of groups on the grid when using configuration κ_0 . Possibly reasons for this phenomenon will be subject of further studies.

VI. OUTLOOK

Further studies will analyze interactions between κ_0 and the number of groups on the grid. An explanation of the phenomenon that group-reputation discourages both cooperation with the in-group and defection with out-groups (model M_2) is of great interest.

Our work can be extended in many ways. Experiments with simulated reputation by introducing fading memory and examining the influence of noisy interactions with both models will be topics of future research.

The field of social simulations and agent societies for exploration and understanding of social processes by means of computer simulation are discussed in other communities, e. g. in sociology. For example, Edmonds (2006) presents an evolutionary simulation where the presence of *tags* and an inbuilt specialization in terms of skills result in the development of *symbiotic* sharing within groups of individuals with similar tags. Concepts from sociology can be used as an inspiration for further research.

REFERENCES

- Axelrod, R. (1984). *The Evolution of Cooperation*. New York, USA: Basic Books.
- Axelrod, R. (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton, USA: Princeton University Press.
- Axelrod, R. & Hammond, R.-A. (2003). The evolution of ethnocentric behavior. In J.-Q. Hill & J.-E. Leighley (Eds.), *Midwest Political Science Convention* Chicago: Midwest Political Science Association.
- Chess, D. M. (1988). Simulating the evolution of behavior: the iterated prisoners' dilemma problem. *Complex systems*, 2(6), 663–670.
- Edmonds, B. (2006). The emergence of symbiotic groups resulting from skill-differentiation and tags. *Journal of Artificial Societies and Social Simulation*, 9(1).
- Fogel, D. B. (1993). Evolving behaviors in the iterated prisoner's dilemma. *Evolutionary Computation*, 1, 77–97.
- Marler, P. & Evans, C. S. (1996). Bird calls: just emotional displays or something more? *Ibis*, 138, 26–33.
- Nowak, M. A. & Sigmund, K. (1998). Evolution of indirect reciprocity by image scoring, the dynamics of indirect reciprocity. *Nature*, 393, 573–577.
- PG 474 (2005). Ipd-dortmund homepage. <http://ls11-www.cs.uni-dortmund.de/people/pg474/>. Cited: 24.1.2005.
- Schenk, S. (1995). *Evolution kooperativen Verhaltens*. Wiesbaden, Germany: Gabler.
- Therneau, T. M. & Atkinson, E. J. (1997). *An Introduction to Recursive Partitioning Using the RPART Routines*. Technical Report 61, Department of Health Science Research, Mayo Clinic, Rochester NY.
- Yao, X. & Darwen, P.-J. (2000). How important is your reputation in a multi-agent environment. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 4(3), 191–200.
- Zahavi, A. & Zahavi, A. (1997). *The Handicap Principle : A Missing Piece of Darwin's Puzzle*. New York: Oxford University Press.

D. Literaturverzeichnis

- [AH03] AXELROD, R. ; HAMMOND, R.-A.: The Evolution of Ethnocentric Behavior. In: *Midwest Political Science Convention*. Chicago, IL, USA : the University of Wisconsin Press, April 2003
- [Ase96] ASENDORPF, J.: *Psychologie der Persönlichkeit*. Berlin, Heidelberg, New York : Springer, 1996
- [Axe97] AXELROD, R.: *The complexity of cooperation : Agent-based models of competition and collaboration*. Princeton University Press, 1997
- [Axe05] AXELROD, R.: *Die Evolution der Kooperation*. Studienausgabe 6.Auflage. München : R.Oldenbourg, 2005
- [B⁺06] BARANSKI, B. u. a.: *High-Order Punishment and the Evolution of Cooperation*. 2006. – <http://www.ipd-dortmund.de/punishment/>. Cited January, 25th 2006
- [BB03] BARTZ-BEIELSTEIN, T.: Experimental Analysis of Evolution Strategies: Overview and Comprehensive Introduction / SFB 531, Universität Dortmund. 2003 (Reihe CI 157/03). – Forschungsbericht. Verfügbar am: 20.02.2006
- [Bäc95] BÄCK, T.: Evolution Strategies: An Alternative Evolutionary Algorithm. In: ALLIOT, J.-M. (Hrsg.) ; LUTTON, E. (Hrsg.) ; RONALD, E. (Hrsg.) ; SCHOENAUER, M. (Hrsg.) ; SNYERS, D. (Hrsg.): *Artificial Evolution*. Berlin, Heidelberg, New York : Springer, 1995 (LNCS 1063), S. 3–20
- [BDM96] BEAUFILS, B. ; DELAHAYE, J. P. ; MATHIEU, P.: Our Meeting with Gradual, A Good Strategy for the Iterated Prisoner’s Dilemma. In: LANGTON, C. G. (Hrsg.) ; SHIMOHARA, K. (Hrsg.): *Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, MA, USA : The MIT Press/Bradford Books, 1996, S. 202–209
- [BNKF98] BANZHAF, W. ; NORDIN, P. ; KELLER, R. E. ; FRANCONI, F. D.: *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. San Francisco : Morgan Kaufmann, 1998
- [CY05] CHONG, S. Y. ; YAO, X.: Behavioral Diversity, Choices and Noise in the Iterated Prisoner’s Dilemma. In: *IEEE Transactions on Evolutionary Computation* 9 (2005), Nr. 6, S. 540–551
- [DH05] DOEBELI, M. ; HAUERT, C.: Models of cooperation based on the Prisoner’s Dilemma and the Snowdrift game. In: *Ecology Letters* 8 (2005), S. 748–766
- [DM95] DELAHAYE, J. P. ; MATHIEU, P.: Complex Strategies in the Iterated Prisoner’s Dilemma. In: ALBERT, A. (Hrsg.): *Chaos and Society* Bd. 29 Université du Québec à Hull, Canada, IOS Press/Presses de l’Université du Québec, 1995, S. 283–292

- [DMB98] DELAHAYE, J. P. ; MATHIEU, P. ; BEAUFILS, B.: Complete classes of strategies for the classical iterated prisoner's dilemma. In: PORTO, V. W. (Hrsg.) ; SARAVANAN, N. (Hrsg.) ; WAAGEN, D. (Hrsg.) ; EIBEN, A. E. (Hrsg.): *Evolutionary Programming VII* Bd. 1447. Springer, 33–41
- [DY95] DARWEN, G. F. ; YAO, X.: On evolving robust strategies for iterated prisoner's dilemma. In: YAO, X. (Hrsg.): *Progress in Evolutionary Computation* Bd. 956. Springer, 276–292
- [Fog93] FOGEL, D. B.: Evolving Behaviors in the Iterated Prisoner's Dilemma. In: *Evolutionary Computation* 1 (1993), S. 77–99
- [HA05] HAMMOND, R.-A. ; AXELROD, R.: *Evolution of Contingent Altruism When Cooperation is Expensive*. September 2005. – <http://www-personal.umich.edu/~axe/>. Cited January, 25th 2006.
- [HC88] HIRSHLEIFER, J. ; COLL, J.: What Strategies Can Support the Evolutionary Emergence of Cooperation? In: *Journal of Conflict Resolution* 32(2) (1988), S. 367–398
- [Hec89] HECKHAUSEN, H.: *Motivation und Handeln*. 2. Auflage. Berlin, Heidelberg, New York : Springer, 1989
- [Hof00] HOFFMANN, R.: *The Ecology of Cooperation. Theory and Decision*. Bd. 50. Amsterdam, The Netherlands : Kluwer Academic Publishers, 2000. – 101–118 S
- [Man01] MANKIW, N. G.: *Grundzüge der Volkswirtschaftslehre*. Stuttgart : Schäffer–Poeschel, 2001
- [MG01] METZ-GÖCKEL, H.: *Kognitionspsychologie Wahrnehmen, Lernen, Informationsverarbeitung, Gedächtnis, Problemlösen*. Vorlesungsskript WS 2001/02 Fachbereich 14 Universität Dortmund, Oktober 2001
- [MG02] METZ-GÖCKEL, H.: *Motivations- und Emotionspsychologie . Grundlagen, Begriffe, Theorien*. Vorlesungsskript WS 2002/03 Fachbereich 14 Universität Dortmund, 2002
- [MNM96] MICHALEWICZ, Z. ; NAZHIYATH, G. ; MICHALEWICZ, M.: A Note on Usefulness of Geometrical Crossover for Numerical optimization Problems. In: FOGEL, L. J. (Hrsg.) ; ANGELINE, P. J. (Hrsg.) ; BÄCK, T. (Hrsg.): *Evolutionary Programming V: Proc. of the Fifth Annual Conf. on Evolutionary Programming*. Cambridge, MA : MIT Press, 1996, S. 305–311
- [Mue88] MUELLER, U.: Optimal Retaliation for Optimal Cooperation. In: *Journal of Conflict Resolution* 31 (1988), S. 692–724
- [Nac92] NACHBAR, J. H.: Evolution in the Finitely Repeated Prisoner's Dilemma. In: *Journal of Economic Behavior and Organization* 19 (1992), S. 307–326
- [NBM94] NOWAK, M. A. ; BONHOEFFER, S. ; MAY, R. M.: Spatial Games and the Maintenance of Cooperation. In: *PNAS* 91 (1994), Nr. 11, 4877–4881. <http://www.pnas.org/cgi/content/abstract/91/11/4877>
- [NS00] NOWAK, M. A. ; SIGMUND, K. ; DIECKMANN, U. (Hrsg.) ; LAW, R. (Hrsg.) ; METZ, J. A. J. (Hrsg.): *Games on Grids*. Cambridge, UK, 2000 (Cambridge Studies in Adaptive Dynamics)

- [Rud99] RUDOLPH, G.: Self-Adaptive Mutations May Lead to Premature Convergence. Dortmund : Fachbereich Informatik, Universität Dortmund, 28 1999 (CI-73/99). – Forschungsbericht.
- [Sch94] SCHWEFEL, H. P.: *Evolution and Optimum Seeking*. New York, NY, USA : John Wiley and Sons, Inc., 1994
- [SHN01] SIGMUND, K. ; HAUERT, C. ; NOWAK, M. A.: Reward and punishment. In: *PNAS* 98 (2001), Nr. 19, 10757-10762. <http://dx.doi.org/10.1073/pnas.161155698>. – DOI 10.1073/pnas.161155698
- [WE00] WITTEN, I. H. ; EIBE, F.: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco : Morgan Kaufmann, 2000
- [Zim88] ZIMBARDO, P. G.: *Psychology and life*. 12th edition. Scott, Foresman (Glenview, Ill.), 1988