Exponentielle untere Schranken zur Lösung infinitärer Auszahlungsspiele und linearer Programme

Oliver Friedmann

Zusammenfassung

Paritätsspiele bilden eine faszinierende Klasse infinitärer Auszahlungsspiele, deren Lösung äquivalent zur Lösung wichtiger Probleme der automatischen Verifikation und Automatentheorie ist. Sie bilden ferner eine sehr natürliche Unterklasse der *Auszahlungsspiele*, die selbst wiederum eine natürliche Teilklasse der *stochastischen Auszahlungsspiele* beschreiben. Aus theoretischer Sicht gehört das Lösen dieser Spiele zu den wenigen Problemen, die in der Komplexitätsklasse $\mathtt{NP} \cap \mathtt{coNP}$ enthalten sind, darüber hinaus kann man zeigen, dass das Lösen in den Klassen $\mathtt{UP} \cap \mathtt{coUP}$ und $\mathtt{PLS}$ enthalten ist. Die Frage, ob eine Klasse dieser Spiele in deterministischer Polynomialzeit gelöst werden kann, gilt als ein wichtiges offenes Problem.

Strategieverbesserung ist eine der wichtigsten algorithmischen Methoden zur Lösung infinitärer Auszahlungsspiele. Sie wird durch eine sogenannte *Verbesserungsregel* parametrisiert, die darüber entscheidet, wie von einer Strategie in der Iteration zur nächsten fortzuschreiten ist. Es ist ein wichtiges offenes Problem, ob es eine Verbesserungsregel gibt, die in einem Polynomialzeitalgorithmus zur Lösung einer der genannten Spieleklassen resultieren würde.

Lineares Programmieren ist eines der wichtigsten Berechnungsprobleme und wird von Wissenschaftlern der Informatik, Mathematik und im Bereich des Operations Research untersucht. Es wurden vermutlich mehr Artikel und Bücher über lineares Programmieren geschrieben als über alle anderen Berechungsprobleme zusammen.

Der *Simplex*- und der *Dual-Simplex*-Algorithmus gehören zu den in der Praxis am häufigsten eingesetzten Algorithmen zur Lösung *linearer Programme*. Simplexverfahren zur Lösung linearer Programme sind eng mit den Strategieverbesserungsalgorithmen verwandt. Ähnlich wie die Strategieverbesserung wird der Simplexalgorithmus durch ein sogenannte *Pivotregel* parametrisiert, die festlegt, wie von einer Basislösung im linearen Programm zur nächsten überzugehen ist. Es ist ein wichtiges offenes Problem, ob es eine Pivotregel gibt, die in einem (starken) Polynomialzeitalgorithmus zur Lösung linearer Programme resultieren würde.

Unser Beitrag zur Strategieverbesserung und zum Simplexverfahren besteht in der Konstruktion exponentieller unterer Schranken für mehrere Verbesserungs- bzw. Pivotregeln. Für jede Verbesserungsregel, die wir

in dieser Arbeit unter die Lupe nehmen, konstruieren wir Zweispieler-*Paritätsspiele*, zu deren Lösung der entsprechend parametrisierte Strategieverbesserungsalgorithmus eine exponentielle Anzahl an Iterationen benötigt. Anschließend übersetzen wir diese Spiele in Einspieler-*Markov-Entscheidungsprozesse*, die wiederum beinahe direkt in konkrete lineare Programme überführt werden können, zu deren Lösung der entsprechende parametrisierte Simplexalgorithmus dieselbe Anzahl an Iterationen benötigt. Zusätzlich zeigen wir, wie sich die unteren Schranken auf expressivere Spieleklassen wie Auszahlungs- und stochastische Auszahlungsspiele übertragen lassen.

Insbesondere zeigen wir exponentielle untere Schranken für die deterministischen *switch all* und *switch best* Verbesserungsregeln zur Lösung von Spielen, für die seit der Einführung von Howards Strategieverbesserungsalgorithmus im Jahre 1960 keine nicht-trivialen unteren Schranken bekannt waren. Darüber hinaus beweisen wir exponentielle untere Schranken für die zwei natürlichsten und am gründlichsten erforschten randomisierten Pivotregeln, nämlich für die *random facet* und die *random edge* Regeln zur Lösung von Spielen und linearen Programmen, für die jahrzehntelang keine nicht-trivialen unteren Schranken bekannt waren. Zusätzlich beweisen wir eine exponentielle untere Schranke für die randomisierte *switch half* Regel zur Lösung von Spielen, die als wichtigste randomisierte multi-switching Regel gilt. Außerdem beweisen wir eine exponentielle untere Schranke für die natürlichste und berühmteste memoisierende Pivotregel von Zadeh zur Lösung von Spielen und linearen Programmen, und lösen damit ein dreißig Jahre lang offenes Problem.

Schließlich beweisen wir exponentielle untere Schranken für zwei andere algorithmische Verfahren zur Lösung von Paritätsspielen, nämlich für den *Model Checking Algorithmus* von Stevens und Stirling sowie für den *rekursiven Algorithmus* von Zielonka.

Synopse

Die Dissertation bietet einen Überblick über unsere untere Schranken an Algorithmen zur Lösung infinitärer Auszahlungsspiele, sowie neue untere Schranken an den Simplexalgorithmus zur Lösung linearer Programme. Insbesondere enthält die Dissertation eine Zusammenfassung der Hauptresultate der folgenden Papiere:

(1) O. Friedmann. Recursive Algorithm for Parity Games requires Exponential Time. In *Theoretical Informatics and Applications, Cambridge Journals, 2011.*

(2) O. Friedmann. An Exponential Lower Bound for the latest Deterministic Strategy Iteration Algorithms. In *Logical Methods in Computer Science, Selected Papers of the Conference LICS 2009.*

(3) O. Friedmann, T. Hansen and U. Zwick. Subexponential Lower Bounds for Randomized Pivoting Rules for Solving Linear Programs. In *Proceedings of*

the *43rd ACM Symposium on Theory of Computing, STOC'11*, San Jose, CA, USA, 2011. Winner of the *Best Paper Award*.

(4) O. Friedmann. A Subexponential Lower Bound for Zadeh's Pivoting Rule for Solving Linear Programs and Games. In *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization, IPCO'11*, New York, NY, USA, 2011. Awarded with *Zadeh's Prize*.

(5) O. Friedmann, T. Hansen and U. Zwick. A Subexponential Lower Bound for the Random Facet Algorithm for Parity Games. In *Proceedings of the Symposium on Discrete Algorithms, SODA'11*, San Francisco, CA, USA, 2011.

(6) O. Friedmann. The Stevens-Stirling-Algorithm for Solving Parity Games Locally Requires Exponential Time. In *International Journal of Foundations of Computer Science, Volume 21, Issue 3*, 2010.

(7) O. Friedmann. An Exponential Lower Bound for the Parity Game Strategy Improvement Algorithm as we know it. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS'09*, Los Angeles, CA, USA, 2009. Winner of the *Kleene Award 2009*.

Ausführliche Zusammenfassung

Die Dissertation befasst sich mit *infinitären Auszahlungsspielen*, die auf der einen Seite ein wichtiges Gebiet der algorithmischen Spieltheorie darstellen und auf der anderen Seite in den Bereichen der modalen Logik und Automatentheorie Anwendung finden. Zusätzlich befasst sich die Arbeit mit *linearer Programmierung*, die wohl eines der wichtigsten Gebiete der konvexen Optimierung darstellt.

Die Dissertation befasst sich hauptsächlich mit dem wichtigsten Algorithmus zur Lösung infinitärer Auszahlungsspiele, nämlich der sog. *Strategieverbesserung*. Hier wird im Sinne der Komplexitätstheorie das worst-case Laufzeitverhalten analysiert. Gleichermaßen wird die worst-case Laufzeit des Simplexverfahrens zur Lösung linearer Programme untersucht.

Infinitäre Auszahlungsspiele Die Arbeit betrachtet eine Vielzahl naheverwandter Klassen von Spielen, die hier zusammenfassend als *infinitäre Auszahlungsspiele* bezeichnet werden. Hierbei handelt es sich um Nullsummenspiele mit perfekter Information, die von einem oder zwei Spielern gespielt werden und darüber hinaus manchmal einen zusätzlichen randomisierten Spieler enthalten, der von der "Natur" kontrolliert wird. Das Spielfeld wird als gerichteter, totaler Graph modelliert, wobei jeder Knoten des Graphs zu einem der Spieler gehört.

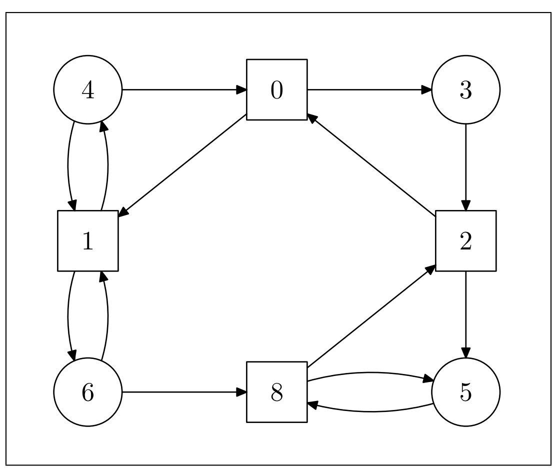
Um eine Partie in so einem Spiel zu spielen, wird ein einzelner Spielstein auf einen Knoten (zum Beispiel auf einen ausgezeichneten Anfangsknoten) des

Graphs gelegt und anhand der ausgehenden Kanten zu einem Nachfolgeknoten gezogen. Der Spieler, zu dem der aktuelle Knoten gehört, entscheidet hierbei, welche ausgehende Kante gewählt werden soll. Falls der aktuelle Knoten zum randomisierten Spieler gehört, so wird eine ausgehende Kante zufällig gewählt. Dieser Prozess wird unendlich lange fortgeführt, woraus sich eine unendliche Knotensequenz ergibt. Nun hängt es von der spezifischen Klasse von Spiel ab, um entscheiden zu können, welche Auszahlung die Spieler erhalten oder welcher Spieler die Partie gewinnt.

Es ist die Aufgabe jedes Spielers die eigene Auszahlung zu maximieren bzw. gegen den anderen Spieler zu gewinnen. Eine *Strategie* eines Spielers ist ein vollständiger Aktionsplan für alle möglichen Situationen, die in einer Partie gegen einen beliebigen Gegner auftreten können. Eine Strategie spezifiziert für jeden Knoten, der zu dem entsprechenden Spieler gehört, welcher Nachfolger auszuwählen ist; dies kann im Allgemeinen von der gesamten *Historie* der Partie bis zu diesem Punkt abhängen. Eine Strategie, die nicht von der Historie abhängig ist, nennt man *positional*.

Alle Spiele, die in dieser Arbeit betrachtet werden, sind *positional determiniert*, d.h. positionale Strategien genügen, um die Entscheidungsprobleme zu beantworten, die mit den Spielen verbunden sind. Dies ist aus vielen Gründen angenehm, da beispielsweise die Anzahl der positionalen Strategien endlich ist (sofern das Spiel endliche Größe hat).

Paritätsspiele sind infinitäre 2-Spieler-Auszahlungsspiele, die auf gerichteten Graphen gespielt werden, deren Knoten mit natürlich Zahlen, genannt *Prioritäten*, beschriftet sind. Die zwei Spieler, genannt *Even* und *Odd*, konstruieren einen unendlichen Pfad im Spielegraph. Even gewinnt, falls die größte Priorität, die unendlich oft in der Partie zu sehen ist, gerade ist. Odd gewinnt andernfalls. Ein Paritätsspiel kann wie folgt visualisiert werden (runde Knoten gehören Spieler Even):



Das Problem, ein Paritätsspiel zu *lösen*, d.h. zu bestimmen, welcher der beiden Spieler eine *Gewinnstrategie* besitzt, ist bekanntermaßen äquivalent zum

Model-Checking-Problem des modalen μ-Kalküls [EJ91, EJS93, Sti95, GTW02]. Darüber hinaus wird die algorithmische Essenz des Paritätsspiellösens zur Lösung vielfältiger Probleme in der Computer-unterstützten Verifikation, nämlich zur Überprüfung auf Validität in Zeit-verzweigenden Logiken [FL10, FLL10] sowie in der Kontrollersynthese [VAW03] eingesetzt.

Paritätsspiele bilden eine sehr spezielle Unterklassen der sog. *Mean Payoff Spiele* [Pur95, EM79, GKK88, ZP96], die selbst eine Unterklasse der sog. *Discounted Payoff Spiele* bilden, die wiederum eine sehr spezielle Unterklasse der rundenbasierten *stochastischen Spiele* [Con92, AM09] bilden. Allgemeinere *stochastische Spiele* wurden zuvor bereits von Shapley [Sha53] untersucht.

Eine weitere, überaus wichtige Klasse infinitärer "Auszahlungspiele" stellen die sog. *Markoventscheidungsprozesse* (nach Andrey Markov) dar, die ein mathematisches Modell zur sequentiellen Entscheidungsfindung in unsicheren Situationen beschreibt. Das Interesse an Markoventscheidungsprozessen begann mit der bahnbrechenden Arbeit von Bellman [Bel57]. Markoventscheidungsprozesse können als spezielle Unterklasse der rundenbasierten stochastischen Spiele aufgefasst werden, in der nur ein Spieler tatsächlich genutzt wird. Markoventscheidungsprozesse haben einen starken Bezug zur Praxis, beispielsweise in der Robotik, der automatisierten Kontrolle, der Ökonomie, im Operations Research sowie in der künstlichen Intelligenz.

Paritäts- und verwandte, ausdrucksstärkere Spieleklassen wie Payoff oder stochastische Spiele, sind bereits für sich alleine genommen ein hochinteressantes Thema aus komplexitätstheoretischer Sicht. Obwohl bekannt ist, dass die Entscheidungsprobleme, die mit diesen Spielen verbunden sind, zu $\texttt{NP} \cap \texttt{coNP}$ [EJS93, Pur95], ja sogar zu $\texttt{UP} \cap \texttt{coUP}$ [Jur98, ZP96], sowie zu $\texttt{PLS}$ [BM08] gehören, ist es ein entscheidendes, offenes Problem, ob eine dieser Spieleklassen in deterministischer Polynomialzeit gelöst werden kann. Auf der anderen Seite ist bekannt, dass Markoventscheidungsprozesse durch spezielle Verfahren aus dem linearen Programmieren bereits in Polynomialzeit gelöst werden können.

Vergleiche dazu Abbildung 1 für einen Überblick über die wichtigsten Beziehungen zwischen den einzelnen Spieleklassen. Es ist leicht zu beobachten, dass Zweispielerspiele die Schwierigkeit, einen Polynomialzeitalgorithmus zur Lösung zu finden, offenbar deutlich erhöhen.

Eine Vielzahl von Algorithmen zur Lösung von Paritätsspielen sind mittlerweile bekannt. Die wichtigsten deterministischen Verfahren sind wohl der rekursive Algorithmus von Zielonka [Zie98], der lokale μ-Kalkül Model-Checker von Stevens und Stirling [SS98], Jurdzińskis *Small Progress Measures*-Algorithmus [Jur00], der subexponentielle Algorithmus von Jurdziński, Paterson und Zwick [JPZ06] mit einer so genannten "big-step" Variante von Schewe [Sch07], sowie mannigfaltige Varianten der Strategieverbesserung (siehe unten), welches die einzige Methode darstellt, die sich auch auf die anderen Spieleklassen übertragen lässt.

Diese Vielfalt ist vermutlich eher der theoretischen Schwierigkeit geschuldet, die es bereitet, die Frage nach der Polynomialzeitlösbarkeit zu beantworten, als praktischen Überlegungen. Die zur Zeit beste bekannte obere Schranke an eine deterministische Lösung von Paritätsspielen ist $\mathcal{O}(e \cdot n^{\frac{1}{3}p})$ mittels Schewes "big-

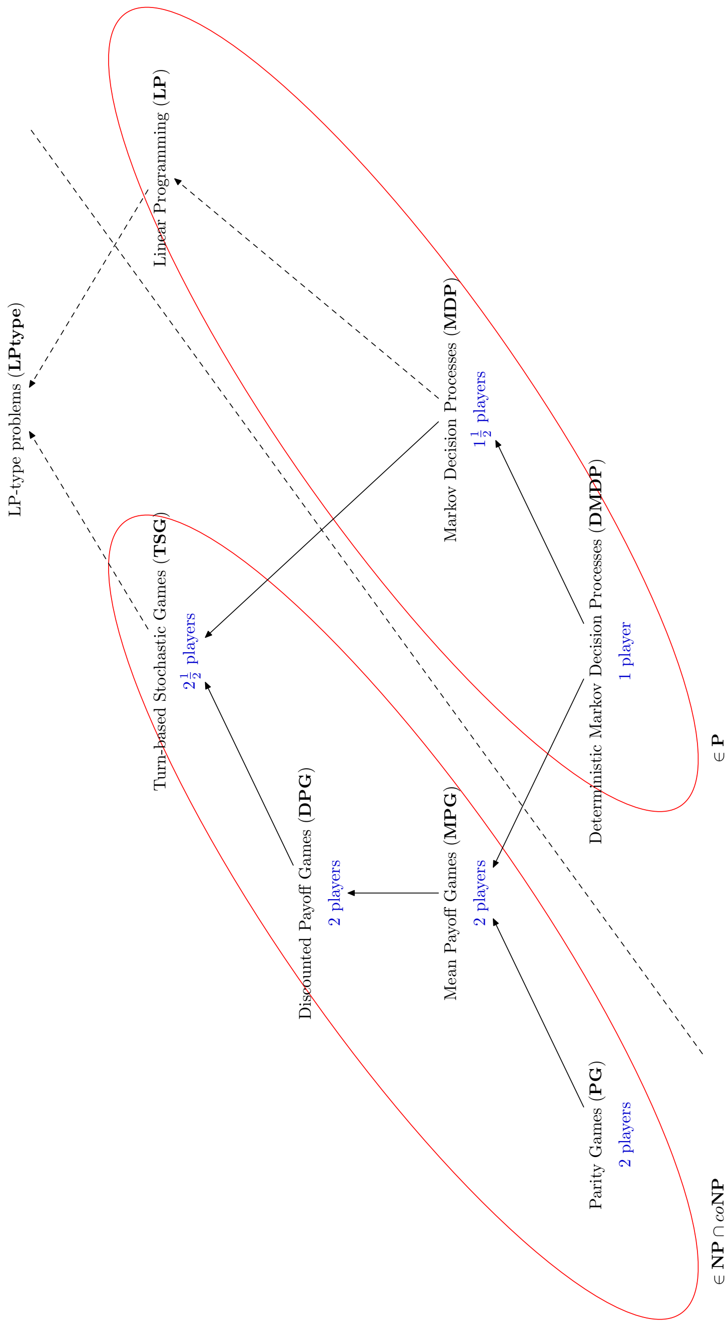


Abbildung 1: Reduktionen und Komplexität

step"-Algorithmus [Sch07], wobei e die Anzahl der Kanten, n die Anzahl der Knoten und p die Anzahl der verschiedenen Prioritäten im Spiel darstellt.

Strategieverbesserung Die *Strategieverbesserung* oder *Strategieiteration* ist ein sehr umfassender Ansatz, der als Lösungsverfahren für infinitäre Auszahlungsspiele und verwandte Probleme angewandt werden kann. Er wurde von Howard [How60] im Jahre 1960 zur Lösung von Markoventscheidungsprozessen eingeführt und wurde von Hoffman und Karp im Jahre 1966 zur Lösung nichtterminierender stochastischer Spiele [HK66] angepasst. Ein paar Jahrzehnte später passte Condon den Algorithmus zur Lösung rundenbasierter stochastischer Spiele [Con92] an; Puri, Zwick und Paterson verwendeten die Methode zur Lösung von Discounted und Mean Payoff Spielen [Pur95, ZP96]. Schließlich formulierten Jurdziński und Vöge eine *diskrete* Variante der Strategieverbesserung zur Lösung von Paritätsspielen [VJ00].

Die Eleganz der Strategieiteration liegt in der Einfachheit ihrer abstrakten Beschreibung begründet. Sie basiert auf einer (Fixpunkt–)Iteration über eine spezielle, endliche Unterklasse der Strategien des ersten Spielers. In jeder Iteration wird die aktuelle Strategie auf eine sog. *Bewertung* abgebildet. Die Bewertung einer Strategie erlaubt es zu entscheiden, *ob* die Strategie bereits *optimal* für den ersten Spieler ist und falls nicht, *wie* die Strategie verändert werden kann, um eine *bessere* zu erhalten. Eine besonders ansprechende Eigenschaft der Strategieverbesserung ist, dass Bewertungen effizient berechnet werden können. Um nun eine optimale Strategie zu finden – die es erlaubt, eine Lösung für das Spiel abzuleiten –, wird das folgende algorithmische Schema angewendet, wobei mit einer beliebigen Strategie σ gestartet wird:

Algorithm 1 Strategieverbesserung

```
while σ ist nicht optimal do
    σ ← Verbessere(σ)
end while
```

Tatsächlich beschreibt die Strategieverbesserungstechnik eine ganze Klasse von Algorithmen, da im Allgemeinen mehr als ein verbesserter Strategiekandidat existiert, mit dem die Iteration fortgesetzt werden kann. Die Methode zur Auswahl der Nachfolgestrategie wird dementsprechend *Verbesserungsregel* genannt. Unter der Annahme, dass nur effiziente Verbesserungsregeln betrachtet werden, lässt sich die Zeitkomplexität der Strategieverbesserung im Prinzip durch die Anzahl der benötigten Iterationen beschreiben, da eine einzelne Iteration in deterministischer Polynomialzeit ausgeführt werden kann.

Somit stellt sich die Frage, ob jede Verbesserungsregel immer zu einer kleinen Anzahl an Iteration führen sollte. Es ist nicht sonderlich überraschend, dass dem nicht so ist. Ein Beispiel ist bereits seit einiger Zeit bekannt, wonach eine absichtlich schlecht gewählte deterministische Verbesserungsregel eine exponentielle Anzahl an Iterationen erzeugen kann [BV07]. Auf der anderen Seiten stellt sich die Frage, ob es überhaupt theoretisch möglich ist, dass es eine Verbesserungsregel geben könnte, die immer für eine kleine Anzahl an Iterationen sorgen würde. Hier ist die Antwort positiv und der einfache Beweis dazu ist wohlbekannt. Allerdings zeigt der Beweis keine Hinweise auf, wie eine solche Verbesserungsregel aussehen könnte. Oder anders formuliert, die Verbesserungsregel, die aus dem Beweis herausfällt, ist selbst nicht effizient berechenbar.

Eine Vielzahl von sehr unterschiedlichen Verbesserungsregeln wurden bereits vorgestellt. Im Allgemeinen gibt es *deterministische*, *randomisierte* und *memorisierende* Verbesserungsregeln. The wichtigsten Regeln, die in der Literatur erwähnt werden, sind die deterministischen Switch-All [VJ00] und Switch-Best [Sch08], die randomisierten Verbesserungsregeln Switch-Half [MS99], Random-Facet [Kal92, Kal97, MSW96] und Random-Edge (Folklore) sowie die memorisierende Least-Entered [Zad80] Regel. Keine nichttrivialen unteren Schranken an die worst-case Laufzeitkomplexität all dieser Regeln waren bis zu dieser Dissertation bekannt.

Wie sich außerdem in dieser Arbeit gezeigt hat, ist die Strategieverbesserung darüber hinaus nahe mit dem sog. *Simplexverfahren* zur Lösung linearer Programme verwandt.

Lineare Programmierung Die lineare Programmierung ist eines der wichtigsten Gebiete der Optimierungstheorie und wird aktiv beforscht. Viele ökonomische und praktische Aufgaben können als lineare Programme formuliert werden und viele Unterprobleme der Informatik basieren auf der Lösung assoziierter linearer Programme.

Ein lineares Programm beschreibt die Aufgabe, eine gegebene *lineare Zielfunktion* zu maximieren (oder zu minimieren), wobei zusätzliche eine Menge linearer Gleichungen und Ungleichungen erfüllt werden muss. Formal ist ein lineares Programm die *Maximierung* einer Zielfunktion

$$c_1 x_1 + \ldots + c_n x_n$$

bezüglich einer Menge linearer (Un)Gleichungen, genannt *Bedingungen*,

$$\begin{aligned} a_{1,1} x_1 + \ldots + a_{1,n} x_n &= b_1 \\ a_{2,1} x_1 + \ldots + a_{2,n} x_n &= b_2 \\ &\vdots \\ a_{m,1} x_1 + \ldots + a_{m,n} x_n &= b_m \end{aligned}$$

wobei alle *Koeffizienten* c_i, alle b_i, sowie alle Koeffizienten $a_{i,j}$ reelle Zahlen sind.

Es zeigt sich, dass Markoventscheidungsprozesse als lineare Programme formuliert werden können, was auch der Grund dafür ist, dass sich diese Klasse von Spielen in polynomieller Zeit lösen lassen.

Es gibt drei grundlegende Algorithmen zur Lösung linearer Programme. Da ist zunächst der *Simplexalgorithmus*, der von Dantzig [Dan63] im Jahre 1963 vorgestellt wurde. Die genaue Laufzeitkomplexität dieses Algorithmus ist unbekannt und es ist ein wichtiges offenes Problem, diese Frage adäquat zu beantworten. Die anderen beiden Algorithmen zur Lösung linearer Programme, nämlich die sog. *Ellipsoidmethode* nach Khachiyan [Kha79] und die sog. *Interior-Point Methode* nach Karmarkar [Kar84], lösen lineare Programme sogar in Polynomialzeit, jedoch nicht in *starker* Polynomialzeit.

Der Unterschied zwischen *starker* und "normaler" oder *schwacher* Polynomialzeit ist subtil und hängt vom Verfahren zur *Messung* der Größe eines gegebenen linearen Programms ab. Offensichtlich sollten die Anzahl der Variablen und die Anzahl der Bedingung *linear* zur Größe beitragen. Aber wie sollen die Koeffizienten gemessen werden? Klassisch würde deren Magnitude und Präzision ebenfalls zur Größe beitragen, da sich eine bestimmte *Kodierung* der Koeffizienten finden lassen müsste. Falls die Laufzeit eines Algorithmus polynomiell bezüglich jenes Maßes ist, so sprechen wir von einem schwachen Polynomialzeitalgorithmus. Falls die Laufzeit hingegen polynomiell in der Anzahl der Variablen und Bedingungen ist (und die Koeffizienten gar nicht betrachtet), so sprechen wir von einem starken Polynomialzeitalgorithmus.

Obwohl bis jetzt nicht einmal geklärt ist, ob das Simplexverfahren tatsächlich ein Polynomialzeitalgorithmus sein könnte, so hat das Verfahren zumindest das Potenzial ein starker Polynomialzeitalgorithmus zu sein, was vermutlich auch der Grund dafür ist, dass sich noch so viele Forscher für dieses Verfahren interessieren.

Simplex Algorithmus Der *Simplexalgorithmus* basiert auf der Beobachtung, dass der Raum der Punkte, die alle Bedingungen erfüllen, im Prinzip ein konvexes Polytop beschreibt (wenn man von ein paar Spezialfällen absieht) und dass die Zielfunktion einen optimalen Wert auf einer der Ecken annimmt. Folgerichtig ist die Idee der Methode, mit einer beliebigen Ecke zu starten, anschließend zu überprüfen, ob die Zielfunktion auf dieser Ecke bereits optimal ist, und falls nicht, zu einer benachbarten Ecke mit besserem Zielfunktionswert zu wechseln.

Ähnlich der Strategieverbesserung wird die Simplexmethode mit einer sog. *Pivotregel* parametrisiert, die bestimmt, welche erlaubte Nachbarecke auszuwählen ist. Gleichermaßen hängt die Laufzeitkomplexität des Simplexalgorithmus hauptsächlich von der Anzahl der *Pivotschritte*, d.h. von der Anzahl der besuchten Ecken ab, da alle anderen Operationen in (starker) Polynomialzeit ausgeführt werden können.

Die Frage, ob jede Pivotregel immer nur eine geringe Anzahl von Pivotschritten benötigen würde, um eine optimale Ecke zu finden, konnte von Klee und Minty [KM72] bereits kurz nachdem Dantzig den Simplexalgorithmus vorgestellt hatte, verneint werden. Im Gegensatz zur Strategieverbesserung ist es ein offenes Problem, ob es überhaupt theoretisch möglich sein könnte, eine kleine Anzahl an Pivotschritten zusichern zu können. Dies ist bekannt als die *Hirschvermutung* (vgl. dazu z.B. [Dan63], pp. 160,168).

Viele der Verbesserungsregeln der Strategieverbesserung können als Pivotregeln für den Simplexalgorithmus (und umgekehrt) formuliert werden, da die meisten bereits abstrakt genug definiert sind. Genau wie bei der Strategieverbesserung waren bis zu dieser Arbeit keine nicht-trivialen unteren Schranken an die worst-case Laufzeitkomplexität für viele von diesen Regeln bekannt.

Es bleibt die Frage, ob es eine tiefere Verbindung zwischen der Strategieverbesserung zur Lösung infinitärer Auszahlungsspiele und dem Simplexalgorithmus zur Lösung linearer Programme geben könnte. In der Tat wird sich zeigen, dass Markoventscheidungsprozesse das "missing link" darstellen, das die Strategieverbesserung mit der Simplexmethode in einer vernünftigen Art und Weise in Beziehung setzen kann.

Beitrag der Dissertation Die Dissertation beschreibt *exponentielle* (d.h. von der Form $2^{\Omega(n)}$) und *subexponentielle* (d.h. von der Form $2^{\Omega(n^c)}$ für ein $0<c<1$) untere Schranken für alle wesentlichen Verbesserungsregeln der Strategieiteration und alle wesentlichen Pivotregelen des Simplexalgorithmus mit bislang ungeklärtem Komplexitätsstatus.

Zunächst werden die Strategieverbesserung für Paritätsspiele unter die Lupe genommen und explizite Spielefamilien konstruiert, auf denen die verschiedenen Verbesserungsregeln exponentielle bzw. subexponentielle Zeit zur Lösung benötigen.

Genauer werden exponentielle untere Schranken für die deterministischen SWITCH-ALL und SWITCH-BEST Regeln zur Lösung infinitärer Auszahlungsspiele, subexponentielle untere Schranken für die randomisierten RANDOM-EDGE und RANDOM-FACET Regeln zur Lösung von Spielen und linearen Programmen, eine subexponentielle untere Schranke für die randomisierte SWITCH-HALF Regel zur Lösung von Spielen, sowie schließlich eine subexponentielle untere Schranke für Zadehs LEAST-ENTERED Regel zur Lösung von Spielen und linearen Programmen beschrieben.

Der Komplexitätsstatus all dieser Regeln war mehrere Jahrzehnte lang ungelöst. Es bestand bis zuletzt die Hoffnung, dass eine dieser Regeln zumindest Paritätsspiele in polynomieller Zeit hätte lösen können.

Zweitens wird gezeigt, dass sich alle Resultate für Paritätsspiele auf die ausdrucksstärkeren Spieleklassen wie Mean und Discounted Payoff Spiele sowie auf rundenbasierte stochastische Spiele übertragen lassen.

Drittens wird beschrieben, wie die Konstruktionen für Paritätsspiele so als Markoventscheidungsprozesse umgebaut werden können, dass sich dieselben Komplexitätsresultate für Strategieverbesserung auf Markoventscheidungsprozessen ergeben. Dies ist im Allgemeinen vermutlich nicht immer möglich, die speziellen Spiele, die in der Arbeit betrachtet werden, lassen sich jedoch alle übersetzen.

Viertens wird die Beziehung zwischen dem Strategieverbesserungsalgorithmus für Markoventscheidungsprozesse und dem Simplexalgorithmus zur Lösung linearer Programme formalisiert, was es erlaubt, alle anwendbaren unteren Schranken auf den Bereich der linearen Programme zu übertragen. Auch hier waren diese Probleme mehrere Jahrzehnte lang ungelöst. Es ist jedoch zu beachten, dass die Resultate dieser Dissertation keine Aussagen über die Gültigkeit der Hirschvermutung treffen können.

Schließlich werden in der Arbeit noch exponentielle untere Schranken für den Model-Checking-Algorithmus nach Stevens und Stirling, sowie für den rekursiven Algorithmus nach Zielonka zur Lösung von Paritätsspielen gezeigt.

Eingesetzte Techniken Alle Konstruktionen zu unteren Schranken, die in der Dissertation für die Strategieverbesserung und das Simplexverfahren beschrieben werden, basieren auf folgenden Schritten:

1. Zunächst werden Familien von Paritätsspielen konstruiert, die jeweils – für jede Verbesserungsregel gesondert – untere Schranken konstituieren. Hierbei beschränken sich die Konstruktionen auf eine in der Arbeit definierte Teilklasse der Paritätsspiele, die sog. *Senken-Paritätsspiele*.

   Die Strategieverbesserung für Paritätsspiele wird als eine Art deterministisches (da beide Spieler deterministischer Natur sind) Berechnungsmodell aufgefasst, in dem sich verschiedene, funktionale Strukturen implementieren lassen. Alle Konstruktionen beschreiben die Implementierung von verschiedenen Varianten binärer Zähler.

   Das spielentscheidende Gadget, das hier eingesetzt wird, um exponentiell viele Iterationen hervorzurufen, wird *einfacher Zykel* genannt, siehe hierzu Abbildung 2.

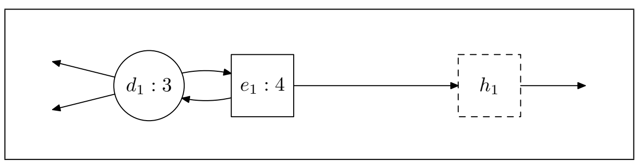
   

   Abbildung 2: Einfacher Zykel

   Die Beweise, dass die Strategieverbesserung auf den Konstruktionen tatsächlich exponentielles bzw. subexponentielles Verhalten zeigt, basieren dann im Wesentlichen darin, zu zeigen, dass die Sequenz der Strategien binäres Zählen nachempfindet, oder zumindest mit hoher Wahrscheinlichkeit "robust genug" binär zählt, sofern randomisierte Verbesserungsregeln betrachtet werden.

   Abbildung 3 zeigt exemplarisch die vollständige Konstruktion eines 3-Bit-Zählers, implementiert in Form eines Paritätsspiels, die exponentiell viele Iterationen zur Lösung benötigt, wenn der Algorithmus mit der Standardverbesserungsregel SWITCH-ALL parametrisiert wird.

2. Die Konstruktionen für Paritätsspiele werden auf die ausdrucksstärkeren Spieleklassen wie Mean und Discounted Payoff Spiele, sowie rundenbasierte stochastische Spiele übertragen.

   Hierzu wird genauer bewiesen, dass sich die Strategieverbesserung auf den sog. Senken-Paritätsspielen exakt gleich verhält, wie die Strategieverbesserung auf den ausdrucksstärkeren Spielklassen, wobei die entsprechenden

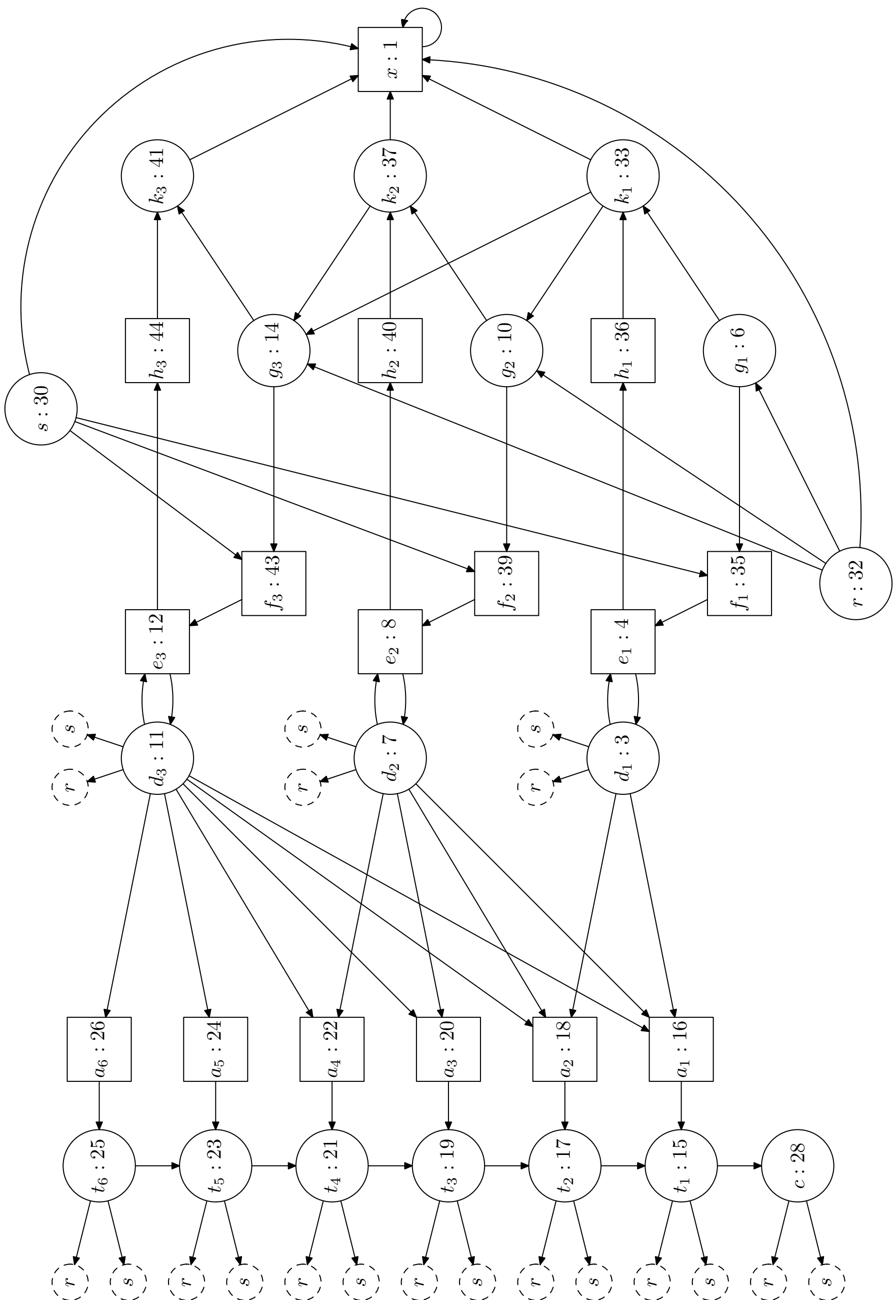


Abbildung 3: Switch-All Lower Bound Spiel G_3

Spiele durch die gängigen Reduktionen von Paritätsspielen konstruiert werden. Diese Beziehung wird unabhängig von der konkreten Verbesserungsregel bewiesen.

3. Die Resultate werden auf Markoventscheidungsprozesse übertragen. Da es (zumindest nach aktuellem Kenntnisstand) keine allgemeine Übersetzung von Paritätsspielen in Markoventscheidungsprozesse gibt, wird die Gültigkeit der Übersetzungen aller Konstruktionen in jedem Einzelfall bewiesen.

   Die einzigen Strukturen, die in den Graphen der Markoventscheidungsprozesse bei der Übersetzung abgeändert werden muss, sind jene Knoten, die von Spieler Odd kontrolliert werden. In den Spielen der Dissertation wird Spieler Odd glücklicherweise nur auf eine Weise eingesetzt, nämlich in Form der einfachen Zyklen und davon leicht abgewandelter Zyklen.

   Diese Struktur kann durch einen Randomisierungsknoten simuliert werden, wobei die Wahrscheinlichkeit dafür, aus dem Zykel hinauszuziehen, exponentiell klein gewählt wird. Vergleiche dazu Abbildung 4 für eine etwas komplizierte Zykelsituation und die Übersetzungsvorschrift von einem Paritätsspiel zu einem Markoventscheidungsprozess.

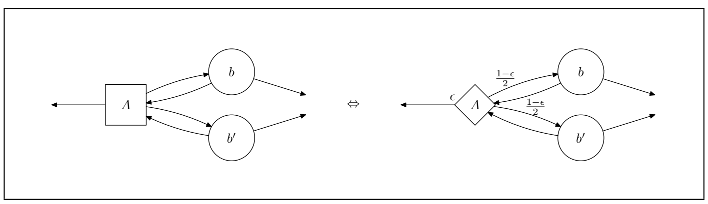


Abbildung 4: Von Paritätsspielen zu MDPs

   Dieses Verfahren wurde durch Fearnleys Arbeit [Fea10] inspiriert und in Zusammenarbeit mit Thomas Dueholm Hansen und Uri Zwick weiterentwickelt.

4. Die Übertragung der Resultate auf den Simplexalgorithmus zur Lösung linearer Programme erfolgt durch den allgemeinen Beweis, dass die Simplexmethode auf linearen Programmen, die durch die Konstruktionen auf Markoventscheidungsprozessen induziert werden, sich genau gleich verhält wie die Strategieverbesserung auf den originalen Markoventscheidungsprozessen.

   Die Bedingungen für optimale Werte (und Potenziale) in einem Markoventscheidungsprozess können als lineares Programm formuliert werden, in dem die Variablen den Werten und die Bedingungen des Programms den Kanten entsprechen.

   Um nun die unteren Schranken für Markoventscheidungsprozesse auf den Simplexalgorithmus zur Lösung linearer Programme zu übertragen, wird gezeigt, dass (1) Basislösungen im induzierten linearen Programm den

Strategien im originalen Spiel entsprechen, und dass (2) angrenzende Basislösungen mit verbesserten Kosten den Strategien entsprechen, die durch eine einzelne Verbesserungskante konstruiert werden. Dies erlaubt es, die unteren Schranken für Markoventscheidungsprozesse direkt auf den Simplexalgorithmus zu übertragen, sofern die entsprechende Verbesserungsregel dementsprechend als Pivotregel formuliert werden kann.

Dieses Verfahren wurde in Zusammenarbeit mit Thomas Dueholm Hansen und Uri Zwick entwickelt.

Weiterführende Forschung Die interessanteste Frage, die unbeantwortet bleibt, ist, ob Paritätsspiele (oder eine der anderen Klassen der infinitären Auszahlungsspiele mit zwei Spielern) in Polynomialzeit lösbar ist. Die Strategieverbesserung scheint noch immer einer der vielversprechendsten Kandidaten für ein Polynomialzeitverfahren zu sein. Nichtsdestotrotz mag es hier notwendig sein, Verbesserungsregeln zu erforschen, die weit über das hinausgehen, was wir heute standardmäßig einsetzen.

Da Paritätsspiele die einfachste Klasse in der Hierarchie der infinitären Zwei-Spieler Auszahlungsspiele darstellen, erscheinen jene Spiele die geeignete Klasse zu sein, um nach einem Polynomialzeitalgorithmus zu suchen. Es ist in der Tat nicht besonders kompliziert zu beweisen, dass viele Verbesserungsregeln Paritätsspiele in Polynomialzeit lösen können, falls Spieler Odd nicht in Strukturen ähnlich der Zyklen, die in der Dissertation aufgezeigt werden, vorkommt. Es könnte hilfreich sein, eine rigorose Analyse jener Strukturen durchzuführen und mit dem gewonnen Wissen möglicherweise eine Verbesserungsregel zu konstruieren, die Paritätsspiele effizient löst.

Bezüglich der Laufzeitkomplexität infinitärer Zweispieler-Spiele wäre es hochinteressant zu sehen, ob sich tiefere Verbindungen zwischen dem Lösen jener Spiele und anderer $\mathtt{NP}$-Probleme herstellen lassen, die weder bekanntermaßen zu $\mathtt{P}$ gehören, noch bewiesenermaßen $\mathtt{NP}$-vollständig sind, wie beispielsweise das Graphenisomorphieproblem oder das Faktorisierungsproblem der natürlichen Zahlen.

Im Bereich des linearen Programmierens umfassen die vermutlich wichtigsten offenen Probleme die Fragen, ob lineare Programme in starker Polynomialzeit gelöst werden können, ob es eine Form der Hirschvermutung gibt, die tatsächlich gilt, und ob es eine Pivotregel gibt, die zu einem Polynomialzeitalgorithmus führen würde.

Es wäre ferner interessant zu untersuchen, ob es möglich ist, Markoventscheidungsprozesse leicht derart abzuändern, dass sie immer noch auf lineares Programmieren reduzierbar bleiben, aber ohne, dass sich noch immer leicht zeigen ließe, dass ihr Durchmesser notwendiger Weise klein sein sollte. Dies würde es theoretisch erlauben, ein Gegenbeispiel zur Hirschvermutung im Bereich der Spiele zu konstruieren, der sich durch diese Dissertation als äußerst hilfreiche Abstraktion zur Konstruktion konkreter linearer Programme erwiesen hat.

Literatur

[AM09] D. Andersson and P. B. Miltersen. The complexity of solving stochastic games on graphs. In *ISAAC '09: Proceedings of the 20th International Symposium on Algorithms and Computation*, pages 112–121, Berlin, Heidelberg, 2009. Springer.

[Bel57] R. E. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[BM08] A. Beckmann and F. Moller. On the complexity of parity games. In *Proceedings of the BCS Conference Visions of Computer Science*, 2008.

[BV07] H. Björklund and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210–229, 2007.

[Con92] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[Dan63] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[EJ91] E. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *Proceedings of the 32nd Symposium on Foundations of Computer Science*, pages 368–377, San Juan, 1991. IEEE.

[EJS93] E. Emerson, C. Jutla, and A. Sistla. On model-checking for fragments of μ-calculus. In *Proceedings of the 5th Conference on CAV, CAV'93*, volume 697 of *LNCS*, pages 385–396. Springer, 1993.

[EM79] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.

[Fea10] J. Fearnley. Exponential lower bounds for policy iteration. *CoRR*, abs/1003.3418, 2010.

[FL10] O. Friedmann and M. Lange. A solver for modal fixpoint logics. *Electronic Notes Theoretical Computer Science*, 262:99–111, May 2010.

[FLL10] O. Friedmann, M. Latte, and M. Lange. A decision procedure for CTL* based on tableaux and automata. In *IJCAR*, pages 331–345, 2010.

[GKK88] V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28:85–91, 1988.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, LNCS. Springer, 2002.

[HK66] A. J. Hofmann and R. M. Karp. On nonterminating stochastic games. *Management Science*, 12(5):359–370, 1966.

[How60] R. Howard. *Dynamic Programming and Markov Processes*. The M.I.T. Press, 1960.

[JPZ06] M. Jurdziński, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm, SODA'06*, pages 117–123. ACM, 2006.

[Jur98] M. Jurdziński. Deciding the winner in parity games is in $\mathtt{UP} \cap \mathtt{coUP}$. *Information Processing Letters*, 68(3):119–124, 1998.

[Jur00] M. Jurdziński. Small progress measures for solving parity games. In H. Reichel and S. Tison, editors, *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS'00*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000.

[Kal92] G. Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proceedings of the 24th STOC*, pages 475–482, 1992.

[Kal97] G. Kalai. Linear programming, the simplex algorithm and simple polytopes. *Mathematical Programming*, 79:217–233, 1997.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, New York, NY, USA, 1984. ACM.

[Kha79] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[KM72] V. Klee and G. L. Minty. How good is the simplex algorithm? *Inequalities*, III:159–179, 1972.

[MS99] Y. Mansour and S. P. Singh. On the complexity of policy iteration. In *Proceedings of the 15th UAI*, pages 401–408, 1999.

[MSW96] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

[Pur95] A. Puri. *Theory of Hybrid Systems and Discrete Event Systems*. PhD thesis, University of California, Berkeley, 1995.

[Sch07] S. Schewe. Solving parity games in big steps. In *Proceedings of the 27th International Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'07*, volume 4855 of *LNCS*, pages 449–460. Springer, 2007.

[Sch08] S. Schewe. An optimal strategy improvement algorithm for solving parity and payoff games. In *Proceedings of the 17th Annual Conference on Computer Science Logic, CSL'08*, volume 5213 of *LNCS*, pages 369–384. Springer, 2008.

[Sha53] L. S. Shapley. Stochastic games. *Proceedings of National Academy of Sciences USA*, 39:1095–1100, 1953.

[SS98] P. Stevens and C. Stirling. Practical model-checking using games. In B. Steffen, editor, *Proceedings of the 4th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'98*, volume 1384 of *LNCS*, pages 85–101. Springer, 1998.

[Sti95] C. Stirling. Local model checking games. In *Proceedings of the 6th Conference on Concurrency Theory, CONCUR'95*, volume 962 of *LNCS*, pages 1–11. Springer, 1995.

[VAW03] A. Vincent, A. Arnold, and I. Walukiewicz. Games for synthesis of controllers with partial observations. *Theoretical Computer Science*, 303(1):7–34, 2003.

[VJ00] J. Vöge and M. Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In *Proceedings of the 12th International Conference on Computer Aided Verification, CAV'00*, volume 1855 of *LNCS*, pages 202–215. Springer, 2000.

[Zad80] N. Zadeh. What is the worst case behaviour of the simplex algorithm? Technical report, Department of Operations Research, Stanford, 1980.

[Zie98] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS*, 200(1–2):135–183, 1998.

[ZP96] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996.