Seminar Algorithmische Spieltheorie und Komplexität, SS 2018
Prof. Dr. Gerhard Woeginger, Björn Tauer

Kommunikationskomplexität der Berechnung eines ϵ-Nash-Gleichgewichts

Michael Krause

RWTH Aachen University, Aachen, Deutschland,
`michael.krause@rwth-aachen.de`.

4. August 2018

Inhaltsverzeichnis

1 Überblick

In dieser Ausarbeitung wird ein Resultat über die Kommunikationskomplexität der Berechnung eines ϵ-Nash-Gleichgewichts bei Bimatrix-Spielen vorgestellt und einige Konzepte seines Beweises erläutert. Die Darstellung basiert auf den Kapiteln 2.1 bis 2.4 von [6].

Im nächsten Abschnitt werden zunächst alle notwendigen Konzepte zum Verständnis des Resultats eingeführt. Anschließend folgt ein Überblick über andere Ergebnisse zur Komplexität von Nash-Gleichgewichten. Schließlich wird das Hauptresultat vorgestellt und die grobe Idee hinter den nötigen Beweisschritten erläutert.

2 Definitionen und Vorarbeit

Diese Arbeit beschäftigt sich mit mathematischen Spielen zwischen zwei Spielern, die unter verschiedenen Spielweisen - auch Strategien genannt - wählen können. Je nach gewählten Strategien erhalten die Spieler unterschiedliche Gewinne. Die gewählten Strategien der Spieler bilden ein Nash-Gleichgewicht, wenn keiner der beiden von seiner Wahl abweichen könnte, um dadurch seinen Gewinn zu vergrößern. Formal werden diese Konzepte wie folgt gefasst:

Definition 1 (Kapitel 1.2.3, 1.2.5 und 1.4 in [6])
Ein Bimatrix-Spiel *ist ein Tupel* (A, B) *von Matrizen* $A, B \in \mathbb{R}^{n\times m}$*, wobei* n *und* m *jeweils die Anzahl der Strategien bezeichnet, die dem sogenannten Zeilenspieler bzw. Spaltenspieler zur Verfügung stehen. Spielt der Zeilenspieler seine* i*-te und der Spaltenspieler seine* j*-te Strategie (für* $1 \leq i \leq n$ *und* $1 \leq j \leq m$*), dann ist der Gewinn des Zeilenspielers* A_{ij} *und der des Spaltenspielers* B_{ij}*.*

Die Spieler können auch Wahrscheinlichkeitsverteilungen über ihre Strategien spielen, also $x \in [0,1]^n, y \in [0,1]^m$ *mit* $\sum_{i=1}^n x_i = \sum_{j=1}^m y_j = 1$*. Der* erwartete Gewinn *beträgt dann* $x^T Ay$ *für den Zeilenspieler und* $x^T By$ *für den Spaltenspieler.*

Ein Nash-Gleichgewicht *ist ein Paar von Wahrscheinlichkeitsverteilungen* $\bar{x}$ *und* $\bar{y}$*, sodass* $\bar{x}^T A\bar{y} \geq x^T A\bar{y}$ *und* $\bar{x}^T B\bar{y} \geq \bar{x}^T By$ *für alle möglichen alternativen Verteilungen* x *und* y *über die Strategien der Spieler.*

Die Anforderungen an ein Nash-Gleichgewicht lassen sich abschwächen. Wenn ein Strategiewechsel für einen Spieler höchstens einen gewissen Vorteil ϵ bringen kann, spricht man von einem ϵ-Nash-Gleichgewicht:

Definition 2 (Definition 1.12 in [6])
Ein ϵ-Nash-Gleichgewicht *eines Bimatrix-Spiels* (A, B) *für ein* $\epsilon > 0$ *ist ein Paar von Wahrscheinlichkeitsverteilungen* $\bar{x}$ *und* $\bar{y}$*, sodass* $\bar{x}^T A\bar{y} \geq x^T A\bar{y} - \epsilon$ *und* $\bar{x}^T B\bar{y} \geq \bar{x}^T By - \epsilon$ *für alle möglichen Verteilungen* x *und* y *über die Strategien der Spieler.*

Eine Teildisziplin der Komplexitätstheorie beschäftigt sich mit der Frage, wie viel Information zwischen Kommunikationsteilnehmern übertragen wird, die ein bestimmtes Problem lösen wollen. Ähnlich wie im Fall von Laufzeit- oder Speicherkomplexität lassen sich auch dafür Schranken herleiten. In diesem Zusammenhang spricht man daher von Kommunikationskomplexität (engl. communication complexity).

Definition 3 (Definition 13.1 in [1])
Sei $\mathbb{B} := \{0,1\}$ *und* $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ *eine Funktion, welche von den Eingaben* $x, y \in \mathbb{B}^n$ *zweier Teilnehmer abhängt. Beide Teilnehmer haben unbegrenzte Rechenkraft, aber keinen Zugriff auf die Eingabe des jeweils anderen Teilnehmers, außer durch Nachrichten, die sie aneinander versenden. Ein* Protokoll *zur Berechnung von* f *beschreibt die zu sendenden Nachrichten zwischen den Teilnehmern auf allen Eingaben* x, y*, sodass beide zuletzt das korrekte Ergebnis* $f(x, y)$ *der Funktion vorliegen haben.*

Die deterministische Kommunikationskomplexität des Protokolls *ist die maximale Länge aller gesendeten Nachrichten über alle möglichen Eingaben. Die* deterministische Kommunikationskomplexität von f *ist die minimale Kommunikationskomplexität über alle Protokolle, die* f *berechnen.*

Es lassen sich auch andere Arten von Kommunikationskomplexität einführen (siehe dazu auch [1, S. 242]): Im Fall der *randomisierten Kommunikationskomplexität* steht beiden Teilnehmern ein gemeinsamer Vorrat an Zufallsbits zur Verfügung. Gezählt wird dann die maximal nötige Anzahl Bits über alle Eingaben und möglichen Zufallsbits, sodass mit einer festen Fehlerwahrscheinlichkeit $< \frac{1}{2}$ das korrekte Ergebnis der Funktion berechnet wird. Alternativ kann die erwartete Anzahl der bei der Berechnung von f übertragenen Bits gemessen werden. Dies ändert die resultierende randomisierte Kommunikationskomplexität allerdings nur um einen konstanten Faktor, genau wie die Wahl der gewünschten Fehlerwahrscheinlichkeit ([5, Kapitel 2.2 und 4.3]).

Im Falle von *nicht-deterministischer Kommunikationskomplexität* können die Teilnehmer auf ein weiteres $z \in \mathbb{B}^*$ zugreifen, das von x und y abhängig sein kann. f gilt dann als berechnet, wenn es ein z gibt, mit dessen Hilfe die beiden Teilnehmer $f(x, y)$ erhalten können. Anschaulich gesprochen ist z ein Zertifikat für $f(x, y)$, dass von einer allwissenden Instanz vorgegeben wird. Die Länge von z zählt dann zur nicht-deterministischen Kommunikationskomplexität dazu.

Ein klassisches Problem, an dem sich Kommunikationskomplexität erläutern lässt, ist DISJUNKTHEIT (engl. disjointness): Die Eingaben $x, y \in \mathbb{B}^n$ der Teilnehmer entsprechen dabei den charakteristischen Vektoren zweier Teilmengen einer endlichen Grundmenge der Größe n. Es ist $f(x, y) = 1$ genau dann, wenn diese Mengen disjunkt sind (wenn es also kein i gibt, sodass $x_i = y_i = 1$). Um diese Funktion zu berechnen, benötigt jedes Protokoll mindestens n Bits an Kommunikation zwischen den Teilnehmern ([1, S.235]):

Satz 4
Die deterministische Kommunikationskomplexität von DISJUNKTHEIT *ist* n.

Beweis
Angenommen, es gäbe ein Protokoll zur Berechnung von f mit Kommunikationskomplexität $\leq n-1$. In diesem Fall sind höchstens 2^{n-1} verschiedene Transkripte der zwischen den Teilnehmern übertragenen Bits möglich. Andererseits gibt es für jedes $x \in \mathbb{B}^n$ ein Eingabepaar $(x, \bar{x})$ und somit 2^n verschiedene derartige Eingabepaare, wobei mit $\bar{x}$ das Komplement zu x bezeichnet wird (also $\bar{x}_i = 1$ genau dann, wenn $x_i = 0$). Es gibt daher $x, x' \in \mathbb{B}^n, x \neq x'$, sodass das Protokoll auf den Eingabepaaren $(x, \bar{x})$ und $(x', \bar{x'})$ das selbe Transkript liefert.

Per Induktion über die Anzahl k der übertragenen Bits lässt sich zeigen, dass dann das Transkript auf dem Eingabepaar $(x, \bar{x'})$ ebenfalls dem Transkript auf $(x, \bar{x})$ entspricht: Für das $k = 1$-te Bit gilt dies, da entweder der Teilnehmer mit Eingabe x sendet (die bei beiden Paaren gleich ist) oder der andere Teilnehmer. Dieser sendet bei Eingabe $(x, \bar{x'})$ jedoch das selbe Bit wie auf $(x', \bar{x'})$, da er schließlich nicht weiß, ob sich die Eingabe des Anderen unterscheidet. Für das $k+1$-te Bit entsprechen nach Induktionsannahme alle bisher gesendeten k Bits dem Transkript auf $(x, \bar{x})$ und $(x', \bar{x'})$. Mit der selben Argumentation wie zuvor ist dann auf Eingabe $(x, \bar{x'})$ auch das $k+1$-te verschickte Bit gleich.

Per Definition gilt $f(x, \bar{x}) = 1$. Allerdings sind x und $\bar{x'}$ nicht disjunkt, da $x \neq x'$, und somit $f(x, \bar{x'}) = 0$. Da die Transkripte auf $(x, \bar{x})$ und $(x, \bar{x'})$ jedoch identisch sind, kann das Protokoll nicht zwischen diesen beiden Ergebnissen unterscheiden.

Es kann daher kein Protokoll für DISJUNKTHEIT mit Kommunikationskomplexität $\leq n-1$ geben. Andererseits lässt sich das Problem sehr einfach mit n Bits an Kommunikation lösen, indem einfach einer der Teilnehmer seine Eingabe vollständig an den Anderen übermittelt ([5, S. 51]). $\square$

Mit etwas weiterer Arbeit lässt sich dieser Beweis auch auf die nicht-deterministische Kommunikationskomplexität von DISJUNKTHEIT übertragen (siehe [5, Kapitel 5.2]). Die randomisierte Kommunikationskomplexität des Problems ist $\Omega(n)$, doch der Beweis dazu ist komplizierter ([5, Kapitel 4.3.4]).

Die letzte Definition betrifft eine Klasse von Funktionen, die einige Male auftauchen werden:

Definition 5
Eine λ-Lipschitz Funktion *auf einer Menge* C *ist ein* $f\colon C \to C$, *sodass für alle* $x, x' \in C$ *gilt, dass*

$$\|f(x) - f(x')\| \leq \lambda \|x - x'\|.$$

3 Andere Resultate zur Komplexität von Nash-Gleichgewichten

Weil Nash-Gleichgewichte eine intuitive Interpretation haben und in der Ökonomie beliebt sind, wären Methoden für eine effiziente Berechnung dieser Gleichgewichte wünschenswert. Ließe sich hingegen beweisen, dass Nash-Gleichgewichte nicht effizient berechenbar sind, so würde das ihre Glaubwürdigkeit als Vorhersage von realen Spielsituationen schwächen (siehe auch [6, Kapitel 1.2.7]).

Für Nullsummenspiele mit zwei Spielern (also Bimatrix-Spiele, bei denen $B = -A$) lassen sich Nash-Gleichgewichte in Polynomialzeit als Lösung eines linearen Programms finden ([6, Claim 1.4]). Dieses Resultat lässt sich jedoch nicht einfach auf generelle Bimatrix-Spiele oder Spiele mit mehr als zwei Spielern erweitern.

Darüber hinaus lassen sich Gleichgewichte in Nullsummenspielen mit zwei Spielern auch durch entkoppelte Dynamik (engl. uncoupled dynamics) erreichen:

Definition 6 (Kapitel 1.3.1 in [6])
Die Spieler eines Bimatrix-Spiels spielen wiederholt Verteilungen über ihre Strategien, also $x_1, x_2, x_3, \ldots$ und $y_1, y_2, y_3, \ldots$ für Zeitschritte $1, 2, 3, \ldots$.

Angenommen, jeder Spieler kennt zu Beginn nur seine eigene Gewinnmatrix und basiert die Strategieverteilung zum Zeitpunkt t nur auf dieser Matrix, sowie den Verteilungen x_i, y_i für $1 \leq i < t$. Dann spricht man von entkoppelter Dynamik.

In Nullsummenspielen mit zwei Spielern gibt es eine Art entkoppelter Dynamik, mit der die Spieler in Zeit logarithmisch in der Anzahl ihrer Strategien ein ϵ-Nash-Gleichgewicht erreichen können. Diese wird geglättetes fiktives Spiel (engl. smooth fictious play) genannt und funktioniert grob, indem die Spieler aus den bisherigen Verteilungen ihres Gegners eine gemittelte Verteilung berechnen und dagegen ihre eigenen Strategien mit Wahrscheinlichkeit je nach erwartbarem Gewinn spielen. Dann liegt den Spielern für ein $\epsilon > 0$ nach $T = \mathcal{O}(\log(n + m)/\epsilon^2)$ Spieldurchläufen ein ϵ-Nash-Gleichgewicht bestehend aus den besagten gemittelten Verteilungen vor ([6, Theorem 1.8]).

Für generelle Bimatrix-Spiele ist eine Methode zur effizienten Berechnung von ϵ-Nash-Gleichgewichten hingegen sehr unwahrscheinlich. Dieses Problem ist für eine Klasse von Suchproblemen namens PPAD vollständig ([6, Theorem 4.4]), die letztendlich dem Auffinden eines Pfadendes in einem Graphen entsprechen (siehe dazu auch Abschnitt 6). Weiterhin gibt es zwar ein QPTAS ([6, Korollar 1.17]), aber kein PTAS für das Problem ([6, Korollar 5.1]). Das bedeutet, dass sich für ein fixes ϵ ein ϵ-Nash-Gleichgewicht in quasi-polynomieller ($2^{\mathcal{O}(\log n)^c}$ für ein festes $c > 0$), aber nicht echt-polynomieller Zeit ($\mathcal{O}(n^c)$ für ein festes $c > 0$) in der Anzahl der Strategien berechnen lässt. Es gibt also einen Algorithmus, der ein Gleichgewicht besser als exponentiell, aber schlechter als polynomiell in n (und möglicherweise auch exponentiell in $\frac{1}{e}$, also sehr schlecht für „gute“ Gleichgewichte) berechnen kann.

In dieser Ausarbeitung geht es jedoch um die Kommunikationskomplexität, nicht die Zeitkomplexität der Berechnung eines ϵ-Nash-Gleichgewichts. Für den Fall von

Nullsummenspielen mit zwei Spielern ergibt sich mit dem oben genannten geglätteten fiktiven Spiel ein Protokoll ([6, Kapitel 1.3.4]): In jedem Spieldurchlauf t ziehen die Spieler - mittels der im randomisierten Szenario verfügbaren Zufallszahlen - gemäß ihrer Verteilungen x_t, y_t eine ihrer Strategien und übertragen diese mit insgesamt $\log(n+m)$ Bits an ihren Gegner. Der kann daraus die oben erwähnte gemittelte Verteilung rekonstruieren. Nach T Runden ist mit hoher Wahrscheinlichkeit ein ϵ-Nash-Gleichgewicht erreicht ([6, Kapitel 1, Fußnote 12]). Die randomisierte Kommunikationskomplexität dieses Protokolls beträgt also $\mathcal{O}(\log^2(n+m)/\epsilon^2)$ Bits. Aus diesem Argument ergibt sich: Wenn es irgendeine Art von entkoppelter Dynamik gibt, die in logarithmischer Zeit in der Anzahl an Strategien zu einem ϵ-Nash-Gleichgewicht konvergiert, dann ergibt sich daraus das oben beschriebene randomisierte Protokoll mit ebenfalls logarithmischer Kommunikationskomplexität.

Kann es auch für den generellen Fall ein Protokoll mit einer geringen Kommunikationskomplexität geben?

Im nicht-deterministischen Szenario ist das tatsächlich der Fall: Weiterhin kennen die Spieler zu Beginn nur ihre eigenen Gewinnmatrizen. Es wird von Bimatrix-Spielen ausgegangen, bei denen beide Spieler die selbe Anzahl N an Strategien haben (ist das nicht der Fall, können zum Beispiel einem der Spieler zusätzliche, „immer schlechte" Strategien zugewiesen werden). Es lässt sich zeigen, dass in solchen Spielen für jedes ϵ ein ϵ-Nash-Gleichgewicht existiert, in dem $\mathcal{O}(\log(N)/\epsilon^2)$ Strategien mit Gleichverteilung gespielt werden ([6, Theorem 1.15]). Jede der N Strategien lässt sich mit $\log(N)$ Bits beschreiben. Somit gibt es ein z der Länge $R = \mathcal{O}(\log^2(N)/\epsilon^2)$, aus dem die zwei Teilnehmer eines Protokolls ein ϵ-Gleichgewicht ohne weitere Kommunikation ablesen können. Dementsprechend ist auch die nicht-deterministische Kommunikationskomplexität in diesem Fall R ([6, Kapitel 1.5.2]).

Gibt es ein ähnliches Resultat für die randomisierte oder deterministische Kommunikationskomplexität von ϵ-Nash-Gleichgewichten?

4 Der Satz von Babichenko und Rubinstein

Die Antwort fällt leider negativ aus. Es lässt sich eine untere Schranke angeben, die polynomiell in der Anzahl der Strategien ist. Dieser erst 2016 von Yakov Babichenko und Aviad Rubinstein bewiesene Satz ist das in den Kapiteln 2.1 bis 2.4 von [6] besprochene Hauptresultat. Er lautet:

Satz 7 (Theorem 2.1 in [6])
Es existiert ein $c > 0$, sodass für kleine $\epsilon > 0$ und große N die randomisierte Kommunikationskomplexität der Berechnung von ϵ-Nash-Gleichgewichten $\Omega(N^c)$ beträgt.

Weil eine in $\mathcal{O}(\log(N))$ Schritten endende entkoppelte Dynamik direkt ein randomisiertes $\mathcal{O}(\log^2(N))$ Protokoll zur Berechnung von ϵ-Nash-Gleichgewichten implizieren würde (siehe Abschnitt 3), kann es gemäß Satz 7 keine derart effiziente Art von entkoppelter Dynamik geben.

Der Satz 7 trifft zwar eine Aussage über randomisierte Kommunikationskomplexität, der in [6] gegebene Beweis gilt aber nur für deterministische Kommunikationskomplexität. Diese Ausarbeitung folgt der Darstellung in [6].

5 Schwierigkeiten beim Beweis

Wie in Abschnitt 2 beschrieben, lässt sich für die Kommunikationskomplexität einiger Probleme eine untere Schranke beweisen. Die erste Idee für einen Beweis von Satz 7 wäre somit eine Reduktion zu einem Problem mit bekannten unteren Schranken wie z.B. DISJUNKTHEIT ([6, Kapitel 2.2]). Es können aber nicht einfach alle positiven Instanzen von DISJUNKTHEIT auf Spiele mit einem ϵ-Nash-Gleichgewicht abgebildet werden, denn laut des Satzes von Nash hat jedes Spiel ein derartiges Gleichgewicht ([6, Satz 1.14]). Hingegen gibt es natürlich negative Instanzen von DISJUNKTHEIT, die nicht abgebildet werden könnten. Der Beweis von Satz 7 darf also nicht direkt von einem Problem ausgehen, das negative Instanzen aufweist.

Wählt man stattdessen ein Kriterium, das Gleichgewichte voneinander unterscheidet (etwa: „In dem Gleichgewicht werden genau vier Strategien mit positiver Wahrscheinlichkeit gespielt“), könnten beispielsweise positive/negative Instanzen von DISJUNKTHEIT auf Spiele abgebildet werden, bei denen alle/kein Gleichgewicht das Kriterium erfüllt. Angenommen, die Reduktion klappt. Die nicht-deterministische Kommunikationskomplexität von DISJUNKTHEIT ist polynomiell in der Größe der Eingabe (siehe Abschnitt 2), während es für ϵ-Nash-Gleichgewichte ein Protokoll gibt, das eine Anzahl Bits logarithmisch in der Größe der Eingabe benötigt (siehe Abschnitt 3). Die Reduktion würde also eine untere Schranke auf ein Problem übertragen, für das bereits ein besseres Protokoll bekannt ist. Eine derartige Reduktion ist demnach unmöglich. Der Beweis von Satz 7 darf also nicht für die nicht-deterministische Kommunikationskomplexität des Problems durchgehen.

6 Die Ideen des Beweises

In aller Kürze verfährt der Beweis von Satz 7 wie folgt: Statt DISJUNKTHEIT wird die Berechnung einer speziellen Art von Fixpunkt auf die Berechnung eines ϵ-Nash-Gleichgewichts reduziert. Weiterhin lässt sich das „Finden eines Pfadendes“ (engl. end of line) auf die Berechnung eines solchen Fixpunkts reduzieren. Schließlich lässt sich eine untere Schranke für die Kommunikationskomplexität von end of line beweisen. Durch die Reduktionen überträgt sich diese untere Schranke auf das ursprüngliche Problem und der Beweis ist abgeschlossen. Für den letzten Schritt des Beweises wird außerdem ein sogenannter Simulationssatz (engl. simulation theorem) verwendet: eine Art, Schranken für Kommunikationskomplexität zu zeigen (siehe dazu Abschnitt 6.3).

Die genannten Reduktionen müssen keine Polynomialzeit-Reduktionen sein, weil beim Modell der Kommunikationskomplexität von unbegrenzter Rechenkraft der Teilnehmer ausgegangen wird. Entscheidend ist die Anzahl übertragener Bits.

In diesem Abschnitt werden die drei an den Reduktionen beteiligten Probleme - Finden eines ϵ-Nash-Gleichgewichts, eines Fixpunkts und eines Pfadendes - zunächst formal gefasst. Es stellt sich heraus, dass sie eng miteinander verwandt sind: Die Berechnung eines ϵ-Nash-Gleichgewichts lässt sich auf die Berechnung eines Fixpunkts reduzieren und diese wiederum auf das Finden eines Pfadendes. Diese Reduktionen sind Thema der folgenden Unterabschnitte und stellen sozusagen die umgekehrte Richtung zum Beweis von Satz 7 dar.

Das Problem, welches letztlich gelöst werden soll, lautet:

Problem 8 (ϵ-NE)
Gegeben sei ein Bimatrix-Spiel durch die Gewinnmatrizen (A, B) und ein $\epsilon > 0$. Finde ein ϵ-Nash-Gleichgewicht des Spiels gemäß Definition 2.

Bei dem zuvor angesprochenen Fixpunktproblem handelt es sich um die Berechnung eines ungefähren Brouwer Fixpunkts, genauer:

Problem 9 (ϵ-BFP)
Gegeben sei ein Orakel für eine stetige Funktion $f\colon C \to C$ über einer kompakten und konvexen Menge $C \subseteq \mathbb{R}^d$. Finde ein $x \in C$ mit $\|f(x) - x\| < \epsilon$. Dann wird x auch ein ϵ-Fixpunkt genannt.

Ein solcher ungefährer Fixpunkt existiert auf jeden Fall, weil nach Brouwers Fixpunktsatz jede stetige Funktion auf kompakten und konvexen Mengen einen Fixpunkt besitzt ([6, Satz 2.2]). Dieses Fixpunktproblem eignet sich für Reduktionen mit ϵ-Nash-Gleichgewichten, weil die Menge aller Verteilungen über Strategien der Spieler ebenfalls kompakt und konvex ist (siehe dazu Unterabschnitt 6.1).

Das dritte Problem betrifft das Auffinden eines Pfadendes in einem Graphen ohne Abzweigungen, der also nur aus Kreisen und Pfaden besteht. Intuitiv ist dieses Problem schwer, weil es zu der offensichtlichen Methode (einem Pfad bis zum Ende zu folgen) keine Alternative zu geben scheint. Formal lautet die Problemstellung:

Problem 10 (EOL)
Gegeben sei ein gerichteter Graph ohne isolierte Knoten, in dem jeder Knoten höchstens eine eingehende und höchstens eine ausgehende Kante hat. Gegeben sei weiterhin ein Startknoten, in den keine Kante eingeht. Der Graph liegt als Orakel vor, dass für jeden Knoten Vor- und Nachfolger ausgibt. Finde einen Endknoten, der also keine ausgehende Kante hat, oder einen anderen Startknoten.

Im Folgenden werden die Probleme zur besseren Lesbarkeit mit ihren englischen Abkürzungen ϵ-NE, ϵ-BFP und EOL (für ϵ-Nash equilibrium, ϵ-Brouwer fixed point und end of line) bezeichnet. In den nächsten beiden Unterabschnitten werden einige klassische Polynomialzeitreduktionen vorgestellt, welche diese drei Probleme in Verhältnis zueinander setzen.

6.1 NE und BFP

In diesem Unterabschnitt wird gezeigt, dass Fixpunkte finden mindestens so schwer ist, wie Gleichgewichte finden. Der Beweis zu Satz 7 verfährt sozusagen in die andere Richtung.

Satz 11 (Kapitel 2.3.2 in [6])
Die Berechnung eines Nash-Gleichgewichts lässt sich auf die Berechnung eines Brouwer Fixpunkts reduzieren.

Beweis
Sei (A, B) ein Bimatrix-Spiel und bezeichne im Folgenden 1 den ersten und 2 den zweiten Spieler. Seien außerdem n und m die Anzahl der Strategien der Spieler 1 und 2. Die Menge der Verteilungen über alle Strategien von Spieler 1 ist dann

$$\Delta_1 = \left\{ x \in \mathbb{R}^n_{\geq 0} \;\middle|\; \sum_{i=1}^{n} x_i = 1 \right\},$$

und analog Δ_2 für Spieler 2. Δ_1 und Δ_2 sind n- bzw. m-Simplexe und damit kompakt und konvex. Dann ist auch ihr Produkt $C = \Delta_1 \times \Delta_2$ kompakt und konvex. Auf dieser Menge wird eine Funktion $f\colon C \to C$ definiert, die jedes Paar von Verteilungen auf ein Paar von möglichen Antworten der Spieler auf ihren Gegner abbildet und deren Fixpunkte den Nash-Gleichgewichten des Bimatrix-Spiels entsprechen. Für $x_1 \in \Delta_1$ und $x_2 \in \Delta_2$ ist f definiert als

$$f(x_1, x_2) = \left(\underset{x'_1 \in \Delta_1}{\arg\max}\, g_1(x'_1, x_2, x_1), \underset{x'_2 \in \Delta_2}{\arg\max}\, g_2(x_1, x'_2, x_2) \right)$$

für Hilfsfunktionen $g_i\colon C \times \Delta_i \to \mathbb{R}, i = 1, 2$. Zum Beispiel ist g_1 für $x_1, x'_1 \in \Delta_1, x_2 \in \Delta_2$ definiert als

$$g_1(x'_1, x_2, x_1) = x'^T_1 A x_2 - \|x'_1 - x_1\|_2^2$$

und g_2 analog. g_i entspricht dem erwarteten Gewinn von Spieler i, wenn er x'_i spielt (vgl. Definition 1), und einem Kostenterm, der große Änderungen in der Verteilung x'_i von Spieler i gegenüber der Verteilung x_i bestraft.

Das Maximum von f ist eindeutig, da die g_i streng konkav in x'_i sind (der erwartete Gewinn ist eine lineare Abbildung, von der ein quadratischer und daher konvexer Term abgezogen wird, siehe Anhang A für die formale Herleitung). Weiterhin ist f auch stetig, da laut dem Maximumssatz die Maximierer einer konkaven Funktion, die stetig in all ihren Parametern ist, über einer konvexen Menge eine stetige Funktion bilden ([7, Korollar 9.20]). Damit sind die Anforderungen von Brouwers Fixpunktsatz erfüllt.

Es gilt nun: $x = (x_1, x_2) \in C$ ist ein Fixpunkt von f genau dann, wenn das Strategiepaar (x_1, x_2) ein Nash-Gleichgewicht des ursprünglichen Spiels bildet. Ist (x_1, x_2) ein Gleichgewicht, dann können die Spieler ihren erwarteten Gewinn nicht unabhängig von dem anderen erhöhen. Demnach sind diese Verteilungen auch Maximierer von g_i und (x_1, x_2) ein Fixpunkt von f.

Sei nun hingegen (x_1, x_2) kein Gleichgewicht und x_1' eine Verteilung, die den Gewinn von Spieler 1 erhöht (das Argument verläuft analog für x_2'). Dann gibt es ein $\delta > 0$ mit

$$\begin{aligned}
& x_1'^T A x_2 - \delta \|x_1' - x_1\|_2^2 > x_1^T A x_2 \\
\Leftrightarrow \quad & \delta x_1'^T A x_2 - \delta^2 \|x_1' - x_1\|_2^2 > \delta x_1^T A x_2 \\
\Leftrightarrow \quad & \delta x_1'^T A x_2 - \|\delta x_1' - \delta x_1\|_2^2 > \delta x_1^T A x_2 \\
\Leftrightarrow \quad & -\delta x_1^T A x_2 + \delta x_1'^T A x_2 - \|\delta x_1' - (1 + \delta - 1) x_1\|_2^2 > 0 \\
\Leftrightarrow \quad & (1-\delta) x_1^T A x_2 + \delta x_1'^T A x_2 - \|(1-\delta) x_1 + \delta x_1' - x_1\|_2^2 > x_1^T A x_2 \\
\Leftrightarrow \quad & ((1-\delta) x_1 + \delta x_1')^T A x_2 - \|(1-\delta) x_1 + \delta x_1' - x_1\|_2^2 > x_1^T A x_2 - \|x_1 - x_1\|_2^2 \\
\Leftrightarrow \quad & g_1((1-\delta) x_1 + \delta x_1', x_2, x_1) > g_1(x_1, x_2, x_1)
\end{aligned}$$

Also ist (x_1, x_2) kein Fixpunkt von f, da x_1 kein Maximierer von $g_1(\cdot, x_2, x_1)$ ist.

Die Reduktion kann in Polynomialzeit durchgeführt werden, da es sich bei f um eine Funktion handelt, die aus zwei konvexen quadratischen Programmen besteht (g_i sind konkav und quadratisch in x_i' und sollen über einer kompakten und konvexen Menge maximiert werden; damit entsprechen sie der Definition eines konvexen quadratischen Programms). Die Optima lassen sich daher in Polynomialzeit finden ([2, Kapitel 1.4.2]). □

Zusammen mit Brouwers Fixpunktsatz, gemäß dem immer ein Fixpunkt existiert, ergibt sich aus Satz 11 auch der Satz von Nash: jedes Bimatrix-Spiel hat ein Nash-Gleichgewicht, welches mit der obigen Reduktion berechnet werden kann.

Mit der selben Konstruktion lässt sich zeigen, dass ϵ-Nash-Gleichgewichte auch $\mathcal{O}(\epsilon)$-Fixpunkten in der l_∞-Norm entsprechen:

Sei (x_1, x_2) ein ϵ-Nash-Gleichgewicht und x_1' der Maximierer von $g_1(\cdot, x_2, x_1)$, x_2' der Maximierer von $g_2(x_1, \cdot, x_2)$. Wenn $x_1' = x_1$ und $x_2' = x_2$, dann ist (x_1, x_2) auch ein ϵ-Fixpunkt. Angenommen, stattdessen $x_1' \neq x_1$ (das Argument für x_2 ist analog). Dann ist

$$g_1(x_1', x_2, x_1) > g_1(x_1, x_2, x_1) \Leftrightarrow x_1'^T A x_2 - \|x_1' - x_1\|_2^2 > x_1^T A x_2,$$

aber da $x_1'^T A x_2 - \epsilon \leq x_1^T A x_2$, da (x_1, x_2) ein ϵ-Nash-Gleichgewicht, muss

$$\|x_1' - x_1\|_2^2 < \epsilon \Leftrightarrow \|x_1' - x_1\|_2 < \sqrt{\epsilon} \Rightarrow \|x_1' - x_1\|_\infty < \mathcal{O}(\epsilon).$$

Sei nun (x_1, x_2) kein ϵ-Nash-Gleichgewicht und x_1', x_2' wie zuvor. Es gibt ein $\bar{x_1}$ mit $\bar{x_1}^T A x_2 - \epsilon > x_1^T A x_2$ (oder ein entsprechendes $\bar{x_2}$, das Argument ist analog), demnach für ein $\delta > 0$:

$$\epsilon < \bar{x_1}^T A x_2 - \delta \|\bar{x_1} - x_1\|_2^2 - x_1^T A x_2$$

Mit den Umformungen aus dem Beweis zu Satz 11 ergibt sich

$$\begin{aligned}
\delta\epsilon < g_1((1-\delta) x_1 + \delta \bar{x_1}, x_2, x_1) - g_1(x_1, x_2, x_1) \leq \\
g_1(x_1', x_2, x_1) - g_1(x_1, x_2, x_1) \leq \lambda \|x_1' - x_1\|,
\end{aligned}$$

wobei g_1 eine λ-Lipschitz Funktion im ersten Argument ist. Somit ist (x_1, x_2) dann auch kein $\frac{\delta\epsilon}{\lambda}$-Fixpunkt in der l_∞-Norm.

6.2 BFP, SPERNER und EOL

Dieser Unterabschnitt behandelt die zweite Reduktion, die im Beweis zu Satz 7 in umgekehrter Richtung (allerdings in einer Variante für zwei Kommunikationsteilnehmer) durchgeführt wird: Pfadenden in einem Graphen ohne Abzweigungen zu finden ist mindestens so schwer, wie Brouwer Fixpunkte zu berechnen.

Die Reduktion geschieht tatsächlich in zwei Schritten: Zunächst wird gezeigt, dass Pfadenden zu finden so schwer ist, wie ein dreifarbiges Unterdreieck in einem speziell gefärbten und unterteiltem Dreieck zu finden. Danach wird die Berechnung eines Fixpunkts auf dieses Problem reduziert.

Problem 12 (SPERNER)
Gegeben sei ein Dreieck A und eine Triangulierung dieses Dreiecks, also eine Unterteilung seiner Fläche in Unterdreiecke. Sei weiterhin jede Ecke von A unterschiedlich gefärbt und jede Ecke eines Unterdreiecks, die auf einer Seite von A liegt, habe eine der beiden Farben der Enden dieser Seite. Für das Problem liegt ein Orakel vor, das für die Koordinaten einer Ecke deren Farbe ausgibt. Finde ein Unterdreieck von A, dessen Ecken drei unterschiedliche Farben haben.

Sperners Lemma ([6, Lemma 2.5]) besagt, dass für jede Instanz von Problem 12 mindestens ein dreifarbiges Unterdreieck existiert.

Satz 13 (Satz 2.1 in [3])
Ein dreifarbiges Unterdreieck lässt sich als das Ende eines Pfades (Problem 10) finden.

Beweis
Zu diesem Zweck wird das Dreieck A mit seiner Unterteilung in einen Graphen umgewandelt: Jedes Unterdreieck wird zu einem Knoten in dem neuen Graphen. Nun werden zwei der drei Farben von A ausgewählt: Zwei Knoten werden miteinander verbunden, wenn sich die dazugehörigen Dreiecke eine Seite teilen, welche die beiden ausgewählten Farben hat; die Kanten werden einheitlich gerichtet (z.B: „Die Ecke mit Farbe 1 liegt immer links von der Richtung der Kante“).

Durch Einfügen weiterer Verbindungen an derjenigen Seite von A, deren Ecken die beiden ausgewählten Farben haben, bleibt zuletzt genau ein Unterdreieck an dieser Seite des Dreiecks übrig (siehe [3, Abbildung 2]). Es wird ein weiterer Knoten X in den Graphen eingefügt und mit dem Unterdreieck an dieser Seite verbunden.

Mit dieser Konstruktion stellt jeder Knoten, der nicht gleich X und zu genau einem anderen Knoten verbunden ist, ein dreifarbiges Unterdreieck dar. Ein solcher Knoten existiert, weil schon X mit nur einem Knoten verbunden ist und die Anzahl der Knoten mit ungeradem Grad in einem Graphen gerade sein muss. Um ein dreifarbiges Unterdreieck zu finden wird also der beschriebene Graph konstruiert und darauf EOL mit X als Startknoten gelöst.

Damit ist die Reduktion komplett. Ein Aufruf des Orakels für den End of Line Graphen lässt sich durch eine konstante Zahl an Abfragen an das Orakel für A

beantworten: Es müssen nur die Ecken desjenigen Dreiecks ausgewertet werden, welches der abgefragte Knoten repräsentiert. Mit einer geeigneten Nummerierung der Knoten im Graphen sowie der Dreiecke in der Triangulierung von A ist die Reduktion dann in Polynomialzeit möglich. □

Weiterhin lässt sich zeigen, dass die Berechnung eines Fixpunkts nicht schwerer ist als das Finden eines dreifarbigen Unterdreiecks. Zusammen mit dem vorigen Satz ist die Reduktion damit komplett.

Satz 14 (Kapitel 2.4 in [6])
Ein $\mathcal{O}(\epsilon)$-Brouwer Fixpunkt lässt sich als dreifarbiges Unterdreieck wie in Problem 12 darstellen.

Beweis (Skizze)
Der Beweis hier gilt nur für λ-Lipschitz (bezüglich der l_2-Norm) Funktionen $f\colon C \to C$ über Dreiecken. Die Menge C ist ein Dreieck, das zwischen den 1-Achsenabschnitten der Koordinatenachsen des dreidimensionalen Koordinatensystems aufgespannt wird:

$$C = \{x \in \mathbb{R}^3_{\geq 0} \mid x_1 + x_2 + x_3 = 1\}$$

Der Beweis lässt sich auch auf andere Fälle erweitern, siehe Kapitel 2.4 in [6].

Sei nun ϵ für die gewünschte Nährung des Brouwer Fixpunkts und λ so gegeben, dass f eine λ-Lipschitz Funktion ist. Sei weiterhin jeder der drei Koordinatenachsen des $\mathbb{R}^3$ eine Farbe zugewiesen. Außerdem wird C in Unterdreiecke aufgeteilt, deren Seitenlängen höchstens ϵ/λ betragen.

Die Elemente $x = (x_1, x_2, x_3) \in C$ werden wie folgt gefärbt: Wenn $f(x) = x$, dann ist x ein Fixpunkt und das Problem gelöst. Ansonsten erhält x die Farbe einer Achse i, sodass $f(x)_i < x_i$. Dann sind die Bedingungen aus Problem 12 erfüllt: $(1,0,0), (0,1,0)$ und $(0,0,1)$ erhalten unterschiedliche Farben, da sie entweder auf sich selbst - also einen Fixpunkt - abgebildet werden, oder sich in ihrem Bild die Koordinate genau einer Achse verringert haben muss. Andererseits gilt für alle $x \in C$ mit $x_i = 0$ für ein i (also Punkte auf einer Seite des Dreiecks), dass nicht $f(x)_j > x_j$ für beide $j \neq i$, weil schon $\sum_{i=1}^{3} x_i = 1$.

Nun gilt, dass jeder Mittelpunkt $p \in C$ eines dreifarbigen Unterdreiecks einem $\mathcal{O}(\epsilon)$-Brouwer Fixpunkt von f in der l_∞-Norm entspricht: Für jede Koordinate i von $f(p)$ ist $f(p)_i \leq p_i + \mathcal{O}(\epsilon)$. Intuitiv gilt dies, da der Abstand von p zu den Ecken des Unterdreicks höchstens ϵ/λ groß war und unter f nicht größer als ϵ werden kann. Andererseits hat sich die i-Koordinate einer der Ecken verringert, da das Unterdreieck dreifarbig ist. Deshalb kann $f(p)_i$ nicht viel größer als p_i werden.

Exemplarisch lässt sich das für $i = 1$ folgendermaßen zeigen: Angenommen, $f(p)_1 = p_1 + \xi$ für ein $\xi > 0$ und $c \in C$ sei eine Ecke des Unterdreiecks, deren 1-Koordinate unter f reduziert wird. Angenommen außerdem, $f(p)_1 > c_1$ (sonst $\xi < \frac{\epsilon}{\lambda} = c_1 - p_1$). Dann

$$p_1 + \xi - c_1 \leq p_1 + \xi - f(c)_1 = \sqrt{(f(p)_1 - f(c)_1)^2} \leq$$
$$\sqrt{\sum_{i=1}^{3}(f(p)_i - f(c)_i)^2} = \|f(p) - f(c)\|_2 \leq \lambda\|p - c\|_2 \leq \lambda\frac{\epsilon}{\lambda} = \epsilon,$$

da aber $|p_1 - c_1| \leq \frac{\epsilon}{\lambda}$ gilt diese Ungleichung nur für $\xi \leq \epsilon + \epsilon/\lambda$.

Da somit $f(p)_i \leq p_i + \mathcal{O}(\epsilon)$ für alle i, gilt mit $\sum_i f(p)_i = 1$ auch $p_i - \mathcal{O}(\epsilon) \leq f(p)_i$. Damit ist jedoch der Abstand von $f(p)$ zu p in der l_∞-Norm ebenfalls $\mathcal{O}(\epsilon)$. Somit ist die Reduktion vollständig: Eine Instanz von $\mathcal{O}(\epsilon)$-BFP lässt sich lösen, indem die hier beschriebene Sperner-Färbung berechnet und ein dreifarbiges Unterdreieck gefunden wird. Weil nur die Mittelpunkte von Unterdreiecken betrachtet werden müssen, ist die Färbung und damit die Reduktion in einer beschränkten Anzahl von Orakelanfragen berechenbar. □

6.3 Simulationssätze

Wie zu Beginn von Abschnitt 6 erläutert, benötigt der Beweis von Satz 7 neben zwei Reduktionen auch eine untere Schranke für die Kommunikationskomplexität von EOL. Statt diese direkt zu beweisen, wird in [6] zunächst eine untere Schranke für die Abfragekomplexität (engl. query complexity) von EOL gezeigt und diese dann durch einen sogenannten Simulationssatz in eine Schranke für die Kommunikationskomplexität umgewandelt.

Die Abfragekomplexität eines Problems betrifft die Anzahl an Bits in der Eingabe, die zur Lösung des Problems mindestens betrachtet werden müssen. Mit „Algorithmus" ist deswegen in der folgenden Definition ein vereinfachtes Berechnungsmodell gemeint: Entscheidungsbäume, die binäre Entscheidungen auf Grund der Bits in der Eingabe treffen (siehe auch [1, Kapitel 12.1]).

Definition 15 (Definition 12.1 in [1])
Sei $f: \mathbb{B}^n \to \mathbb{B}$ eine Funktion und bezeichne als die Kosten *eines Algorithmus auf einer Eingabe $x \in \mathbb{B}^n$ die Anzahl an Bits, die der Algorithmus in x betrachtet, um $f(x)$ zu berechnen. Bezeichne weiterhin als* maximale Kosten *eines Algorithmus die höchsten Kosten über alle Eingaben x.*

Die Abfragekomplexität *von f sind dann die niedrigsten maximalen Kosten aller Algorithmen, die f berechnen.*

Simulationssätze werden generell genutzt um untere Schranken für die Abfragekomplexität von f in untere Schranken für die Kommunikationskomplexität einer Funktion f' zu übertragen, wobei f' die selbe Ausgabe wie f berechnet, aber von den Eingaben zweier Teilnehmer abhängt (für Weiteres dazu und eine Liste bekannter Simulationssätze siehe [4]). Im Beweis von Satz 7 wird ein solcher Simulationssatz ([6, Satz 2.7]) auf EOL angewandt, wobei sich die Kosten dann nicht auf Bits sondern auf die Anzahl an Orakelanfragen bezieht. Die Resultierende untere Schranke für die Kommunikationskomplexität von EOL bildet die Basis für die Reduktionen des Beweises.

6.4 Weitere Aspekte des Beweises

Die Details der Beweisschritte werden in den Kapiteln 2.6-2.7 und 3.1-3.2 von [6] beschrieben. Tatsächlich sind die Reduktionen dort nicht direkt die Umkehrrichtung der Ergebnisse aus den vorigen Unterabschnitten, sondern beziehen sich auf Varianten der Probleme aus Abschnitt 6, welche für zwei Kommunikationsteilnehmer definiert werden. Aus der Anwendung eines Simulationssatzes auf EOL entsteht etwa eine EOL-Variante für zwei Teilnehmer ([6, Kapitel 2.7]). Für ϵ-BFP entsteht eine recht komplizierte Familie von Funktionen f, die implizit durch die Eingaben der beiden Teilnehmer definiert ist ([6, Kapitel 3.1.3 und 3.1.4]).

Damit die Kommunikationsteilnehmer überhaupt gemeinsam einen ϵ-Fixpunkt berechnen können, muss es mindestens einen solchen Punkt geben, der sich in einer beschränkten Menge an Bits übertragen lässt. Deshalb sind die im Beweis konstruierten f immer λ-Lipschitz Funktionen. Ein Fixpunkt eines solchen f lässt sich in einer Anzahl Bits polynomiell in der Dimension n der Funktion und $\log(1/\epsilon)$ zu einem ϵ-Fixpunkt kodieren:

Sei $f\colon C \to C$ eine Funktion über einem kompakten $C \subseteq \mathbb{R}^n$. Der Raum $\mathbb{R}^n$ wird zu einem Raster diskretisiert, sodass der Abstand nebeneinander liegender Rasterpunkte $\epsilon/(\lambda+1)$ beträgt. Im Einheitswürfel des $\mathbb{R}^n$ liegen so zum Beispiel $((\lambda+1)/\epsilon)^n$ Rasterpunkte. Jeder Punkt $x \in C$ lässt sich jetzt zum nächstgelegenen Rasterpunkt x' runden, sodass der Abstand zwischen x und x' höchstens $\epsilon/(\lambda+1)$ beträgt. Deshalb gilt für jeden Fixpunkt $f(x) = x$ und dessen nächsten Rasterpunkt x',

$$\|f(x') - x'\| = \|f(x') - x + x - x'\| \leq \|f(x') - x\| + \|x - x'\| = \\ \|f(x') - f(x)\| + \|x - x'\| \leq \lambda\|x' - x\| + \|x - x'\| \leq \epsilon$$

wobei verwendet wird, dass $\|x' - x\| \leq \epsilon/(\lambda+1)$ und f eine λ-Lipschitz Funktion ist. Jeder Rasterpunkt innerhalb des Einheitswürfels lässt sich nun mit $\log(((\lambda+1)/\epsilon)^n) = \mathcal{O}(n\log(1/\epsilon))$ Bits beschreiben. Entsprechend sind alle Rasterpunkte in C in $\mathrm{Poly}(n\log(1/\epsilon))$ Bits beschreibbar, insbesondere auch der ϵ-Fixpunkt x'.

7 Abschluss

In dieser Ausarbeitung wurde ein Resultat vorgestellt, das die Kommunikationskomplexität der Berechnung eines ϵ-Nash-Gleichgewichts polynomiell in der Anzahl der Strategien der Spieler nach unten beschränkt. Insbesondere wird dadurch eine entkoppelte Dynamik, die wie im Fall eines Nullsummenspiels für zwei Spieler in logarithmischer Zeit zum Gleichgewicht konvergiert, unmöglich (siehe Abschnitt 3).

Zusammen mit anderen Resultaten wie der PPAD-Vollständigkeit von ϵ-Nash-Gleichgewichten, erscheint das Nash-Gleichgewicht als „die Lösung“ eines Spiels also aus Komplexitätsgründen zunehmend unrealistisch. Alternativen dazu bieten abgeschwächte Gleichgewichtskonzepte, wie sie etwa im Kapitel II.5 von [6] beschrieben werden.

Anhang A g_i ist streng konkav

Damit die im Beweis von Satz 11 verwendete Funktion f wohldefiniert ist, müssen die Maxima der Funktionen g_i eindeutig sein. Dies ist der Fall, wenn sie streng konkav in dem zu maximierenden Argument x'_i sind. Das wird hier für die Funktion

$$g_1(x'_1, x_2, x_1) = x_1'^T A x_2 - \|x'_1 - x_1\|_2^2$$

gezeigt. Für feste $x_1 \in \Delta_1, x_2 \in \Delta_2$ und beliebige $x'_1, \bar{x}'_1 \in \Delta_1, x'_1 \neq \bar{x}'_1, \lambda \in (0,1)$ soll also gelten:

$$g_1(\lambda x'_1 + (1-\lambda)\bar{x}'_1, x_2, x_1) > \lambda g_1(x'_1, x_2, x_1) + (1-\lambda) g_1(\bar{x}'_1, x_2, x_1)$$

Durch Einsetzen der Definition von g_1 und Umformungen ergibt sich:

$$\begin{aligned}
& g_1(\lambda x'_1 + (1-\lambda)\bar{x}'_1, x_2, x_1) > \lambda g_1(x'_1, x_2, x_1) + (1-\lambda) g_1(\bar{x}'_1, x_2, x_1) \\
\Leftrightarrow\quad & (\lambda x'_1 + (1-\lambda)\bar{x}'_1)^T A x_2 - \left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - x_1\right\|_2^2 > \\
& \lambda x_1'^T A x_2 - \lambda \|x'_1 - x_1\|_2^2 + (1-\lambda)\bar{x}_1'^T A x_2 - (1-\lambda)\left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & \lambda x_1'^T A x_2 + (1-\lambda)\bar{x}_1'^T A x_2 - \left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - x_1\right\|_2^2 > \\
& \lambda x_1'^T A x_2 - \lambda \|x'_1 - x_1\|_2^2 + (1-\lambda)\bar{x}_1'^T A x_2 - (1-\lambda)\left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & -\left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - x_1\right\|_2^2 > -\lambda \|x'_1 - x_1\|_2^2 - (1-\lambda)\left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & \left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - x_1\right\|_2^2 < \lambda \|x'_1 - x_1\|_2^2 + (1-\lambda)\left\|\bar{x}'_1 - x_1\right\|_2^2
\end{aligned}$$

Nun ist

$$\begin{aligned}
\left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - x_1\right\|_2^2 &= \left\|\lambda x'_1 + (1-\lambda)\bar{x}'_1 - (\lambda + 1 - \lambda)x_1\right\|_2^2 = \\
\left\|\lambda x'_1 - \lambda x_1 + (1-\lambda)\bar{x}'_1 - (1-\lambda)x_1\right\|_2^2 &= \left\|\lambda(x'_1 - x_1) + (1-\lambda)(\bar{x}'_1 - x_1)\right\|_2^2 = \\
\|\lambda(x'_1 - x_1)\|_2^2 + \left\|(1-\lambda)(\bar{x}'_1 - x_1)\right\|_2^2 &+ 2\lambda(x'_1 - x_1)^T(1-\lambda)(\bar{x}'_1 - x_1) = \\
\lambda^2\|x'_1 - x_1\|_2^2 + (1-\lambda)^2\left\|\bar{x}'_1 - x_1\right\|_2^2 &+ 2(\lambda - \lambda^2)(x'_1 - x_1)^T(\bar{x}'_1 - x_1)
\end{aligned}$$

und weiteres Umformen der Ungleichung ergibt

$$\begin{aligned}
& 2(\lambda - \lambda^2)(x'_1 - x_1)^T(\bar{x}'_1 - x_1) < (\lambda - \lambda^2)\|x'_1 - x_1\|_2^2 + \\
& \qquad\qquad ((1-\lambda) - (1-\lambda)^2)\left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & 2(\lambda - \lambda^2)(x'_1 - x_1)^T(\bar{x}'_1 - x_1) < (\lambda - \lambda^2)\|x'_1 - x_1\|_2^2 + (\lambda - \lambda^2)\left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & 2(x'_1 - x_1)^T(\bar{x}'_1 - x_1) < \|x'_1 - x_1\|_2^2 + \left\|\bar{x}'_1 - x_1\right\|_2^2 \\
\Leftrightarrow\quad & 0 < ((x'_1 - x_1) - (\bar{x}'_1 - x_1))^2,
\end{aligned}$$

wobei die vorletzte Umformung nutzt, dass $(\lambda - \lambda^2) > 0$, weil $\lambda \in (0,1)$, und die letzte Ungleichung gilt, da $x'_1 \neq \bar{x}'_1$. Damit ist g_1 streng konkav, das Argument für g_2 ist analog.

Literatur

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, 1st edition, 2009.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[3] Paul W. Goldberg. A survey of ppad-completeness for computing nash equilibria. arXiv:1103.2709 [cs.GT], 2011.

[4] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. arXiv:1703.07666 [cs.CC], 2017.

[5] Tim Roughgarden. Communication complexity (for algorithm designers). arXiv:1509.06257 [cs.CC], 2015.

[6] Tim Roughgarden. Complexity theory, game theory, and economics. arXiv:1801.00734 [cs.CC], 2018.

[7] Rangarajan K Sundaram. *A First Course in Optimization Theory.* Cambridge University Press, 1996.