

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK

Bachelorarbeit zum Thema

Komplexität in Congestion Spielen und das Nash-Gleichgewicht

Sally Chau
Matrikelnummer 2536412

Erstprüfer: Prof. Dr. Heiko Röglin
Zweitprüfer: Prof. Dr. Norbert Blum

28. September 2015

Danksagung

An dieser Stelle bedanke ich mich bei meiner Familie und meinen Freunden, die mich auf dem Weg bis zum heutigen Tage begleitet und unterstützt haben. Besonderer Dank gilt hierbei Ramtin Azimi, mit dem ich während der Zeit des Bachelors durch dick und dünn gegangen bin.

Weiterhin bedanke ich mich bei Herrn Prof. Dr. Heiko Röglin, der mir während des gesamten Studiums als Professor, Betreuer und Gutachter zur Seite stand. Außerdem bedanke ich mich bei Herrn Prof. Dr. Norbert Blum, der die Rolle des Zweitgutachters übernommen hat.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig, ohne fremde Hilfe angefertigt worden ist. Inhalte und Passagen, die aus fremden Quellen stammen und direkt oder indirekt übernommen worden sind, wurden als solche kenntlich gemacht. Ferner versichere ich, dass ich keine andere, außer der im Literaturverzeichnis angegebenen Literatur verwendet habe. Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Bonn, den 28. September 2015, Sally Chau

Inhaltsverzeichnis

1	Einleitung	9
2	Das Framework	11
2.1	Spieltheorie	11
2.2	Das Nash-Gleichgewicht	13
2.3	Die Klasse \mathcal{PLS}	15
3	Congestion Spiele und das Nash-Gleichgewicht	17
3.1	Rosenthals Potentialfunktion	17
3.2	Netzwerk Congestion Spiele	22
3.2.1	Allgemeine Netzwerk Congestion Spiele	24
3.2.2	Symmetrische Netzwerk Congestion Spiele	30
4	Fazit	33

Kapitel 1

Einleitung

Congestion Spiele, erstmals eingeführt durch Rosenthal [9], bestehen aus einer Menge von Spielern und einer Menge von Strategien. Eine Strategie entspricht in einem Congestion Spiel einer Menge von Ressourcen, die einem Spieler zur Verfügung stehen. Die Verwendung einer Ressource bringt Kosten mit sich, die davon abhängen, wie viele Spieler eine Ressource belegen. Wird eine Ressource nicht verwendet, verursacht sie keine Kosten. Das Ziel eines Spielers ist es, die Ressourcen möglichst so zu wählen, dass die eigenen Kosten gering bleiben.

Dieses Modell findet auch in der Praxis ihre Anwendung. Beispielsweise lässt sich die Produktionstechnik durch ein Congestion Spiel modellieren. Verschiedene Firmen benötigen Ressourcen in Form von Gütern. Dabei steigt der Preis einer Ressource, wenn die Nachfrage hoch ist. Es stellt sich die Frage, ob sich alternative Ressourcen kostengünstiger für eine Firma auszahlen. Diese Anschauung lässt sich gut auf die Ressource der Rohstoffe anwenden. Je öfter ein Rohstoff, beispielsweise Öl, gebraucht wird, desto geringer wird das Vorkommen, welches sich in einem erhöhten Preis äußert. Sinkt dagegen der Verbrauch und die Nachfrage nach dem Rohstoff, kann es passieren, dass auch der Preis sinkt. Ein weiteres Beispiel für ein Congestion Spiel findet sich in der Routenplanung. Die möglichen Pfade, um von einem Start- zu einem Zielpunkt zu gelangen, bilden die Ressourcen eines Congestion Spiels. Das Ziel eines Spielers ist es, unter der Verwendung der Pfade, möglichst kostengünstig zu reisen. Dabei werden die Kosten der Route durch die Pfadwahl der anderen Spieler beeinflusst.

Rosenthal [9] hat bewiesen, dass jedes Congestion Spiel ein Nash-Gleichgewicht besitzt. In einem Nash-Gleichgewicht hat kein Spieler den Anreiz seine Strategie zu ändern, da sich dadurch kein Mehrwert für ihn ergeben würde. Mithilfe einer Potentialfunktion zeigt Rosenthal, dass dieses in endlich vielen Schritten erreichbar ist. Fabrikant, Papadimitriou und Talwar [3] zeigen, dass das tatsächliche Erreichen eines Gleichgewichtes exponentiell viele Schrit-

te benötigen kann. Weiterhin zeigen sie, dass die Berechnung eines Nash-Gleichgewichtes für Congestion Spiele im Allgemeinen \mathcal{PLS} -vollständig ist. Wir werden die Klasse \mathcal{PLS} , die von Johnson, Papadimitriou und Yannakakis [6] eingeführt wurde, im nächsten Kapitel vorstellen. Ackermann, Röglin und Vöcking [1] erweitern die \mathcal{PLS} -Vollständigkeit der Congestion Spiele auf asymmetrische Netzwerk Congestion Spiele. Für symmetrische Netzwerk Congestion Spiele hingegen, beweisen Fabrikant, Papadimitriou und Talwar [3], dass es möglich ist ein Nash-Gleichgewicht in polynomieller Zeit zu berechnen.

In dieser Arbeit werden wir zunächst die notwendigen Definitionen und Notationen angeben. Hierbei werden wir den Kontext der Spieltheorie ausleuchten und die Definition eines Congestion Spiels angeben. Daraufhin stellen wir die oben genannten Ergebnisse aus der Forschung der Congestion Spiele dar, die für diese Arbeit relevant sind. Hierbei konzentrieren wir uns auf die Komplexität, ein Nash-Gleichgewicht für Congestion Spiele mittels der Minimierung von Rosenthals Potentialfunktion, zu finden. Das Ziel dieser Arbeit wird es sein, diese Ergebnisse zu erweitern, indem wir im nächsten Schritt allgemeine Congestion Spiele untersuchen. Wir zeigen, dass die Minimierung von Rosenthals Potentialfunktion im Allgemeinen stark \mathcal{NP} -schwer ist. Dieses Ergebnis weiten wir auf asymmetrische Netzwerk Congestion Spiele aus.

Kapitel 2

Das Framework

In diesem Kapitel klären wir die notwendigen Definitionen und Notationen, um das benötigte Framework für diese Arbeit zu setzen. Hierzu geben wir einen kurzen Einblick in die Ideen der Spieltheorie, sowie die Definition des Congestion Spiels. Weiterhin beschäftigen wir uns in diesem Kontext mit dem Begriff des Nash-Gleichgewichtes, der in dieser Arbeit oft verwendet wird.

2.1 Spieltheorie

Die *Spieltheorie* ist ein weiter Begriff, der sich nicht nur in der Mathematik und in Wirtschaftswissenschaften wiederfindet, sondern auch in Bereichen der Psychologie, Biologie oder auch den Sozialwissenschaften. Die Wurzeln der Spieltheorie finden sich in der Untersuchung von *Null-Summen-Spielen* des John von Neumanns. Die Spieltheorie beschäftigt sich mit strategischem Handeln und modelliert diese auf logische und mathematische Weise. Hierbei betrachten wir *Spieler*, die sich in einem *Spiel* gegenüberstehen. Häufig betrachtete Spiele sind unter anderem Tik-Tak-Toe, Schach oder auch das Gefangenendilemma. All diese Spiele haben gemein, dass der Gewinn bzw. der Verlust eines Spielers von dem Gewinn bzw. Verlust der anderen Spieler abhängt. Die Spieltheorie untersucht die Ergebnisse eines Spielers in Abhängigkeit von den Strategiekombinationen aller Spieler. Die Besonderheit bei den zuvor genannten Null-Summen-Spielen besteht darin, dass der Gewinn eines Spielers dem Verlust eines anderen Spielers entsprechen muss. Summiert man die Gewinne und Verluste aller Spieler, so erhält man eine Summe von null.

Die Spieltheorie lässt sich in zwei Bereiche einteilen: die *kooperative* und die *nicht-kooperative* Spieltheorie. Letztere beschreibt eine Spielumgebung, in der jeder Spieler egoistisch und intelligent handelt, sodass die eigenen Bedürfnisse und Ziele erreicht werden. Diese Art der Spieltheorie bildet auch den Kontext dieser Arbeit.

Im Folgenden bieten wir die notwendigen Definitionen zur Analyse der *Congestion Spiele*. Hierzu betrachten wir zunächst den allgemeinen Begriff eines Spiels.

Definition 2.1 (Spiel). Ein Spiel \mathcal{G} ist ein Tupel $(\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (U_i)_{i \in \mathcal{N}})$, wobei $\mathcal{N} = \{1, \dots, n\}$ die Menge der Spieler beschreibt, Σ_i den Strategieraum eines Spielers $i \in \mathcal{N}$ und $U_i : \Sigma_1 \times \dots \times \Sigma_n \rightarrow \mathbb{R}^n$ die Auszahlungsfunktion für einen Spieler $i \in \mathcal{N}$ in einem Zustand S . Mit $S = (S_1, \dots, S_n) \in \Sigma_1 \times \dots \times \Sigma_n$ beschreiben wir den Zustand eines Spieles, in dem Spieler i Strategie $S_i \in \Sigma_i$ spielt.

Für einen Spieler i unterscheiden wir die Art seiner Strategie. Anschaulich stellt eine Strategie einen Handlungsplan dar, der vorsieht in welchem Zustand welche Handlung gespielt wird. Hierbei gibt es zwei elementare Strategiearten. Wir sprechen von einer *reinen* Strategie, wenn durchweg eindeutig bestimmt ist, welche Aktion Spieler i ausübt. In einer *gemischten* Strategie dagegen, wählt Spieler i über eine Wahrscheinlichkeitsverteilung eine Strategie aus der Menge seiner Strategien aus. Es ist daher nicht immer eindeutig und von vornherein absehbar, welche Aktion durch Spieler i erfolgt.

Wir definieren den Zustand $S \oplus S'_i := (S_1, \dots, S_{i-1}, S'_i, S_{i+1}, \dots, S_n)$. Die Zustände S und $S \oplus S'_i$ unterscheiden sich darin, dass Spieler i im Zustand $S \oplus S'_i$ die Strategie S'_i anstelle von S_i spielt. Eine Strategie S_i eines Spielers *dominiert* eine Strategie S'_i , falls $U_i(S) \geq U_i(S \oplus S'_i)$.

Wir nennen ein Spiel *symmetrisch*, wenn die Menge der Strategien für alle Spieler i gleich ist, ansonsten nennen wir ein Spiel *asymmetrisch*.

Eine Klasse von Spielen, die in den Rahmen der Definition 2.1 passt, sind *Congestion Spiele*. Für eine Menge von Ressourcen entscheidet jeder Spieler in einem Congestion Spiel, ob er eine bestimmte Ressource für sich beansprucht oder nicht. Hierbei sollen die dabei entstehenden Kosten möglichst gering gehalten werden. Ein Beispiel für ein Congestion Spiel im Alltag ist der Einkauf von Gütern von Firmen in der Industrie. Die Menge der Ressourcen besteht aus den verfügbaren Gütern, wie zum Beispiel Rohstoffen. Die konkurrierenden Firmen werden durch die Spieler im Congestion Spiel modelliert. Als Firma gilt es darüber zu entscheiden, welche und wie viele Güter beansprucht werden sollen. Je mehr Firmen eine Ressource nachfragen, desto höher ist ihr Preis. Das Ziel jeder Firma ist es, die eigenen Kosten gering zu halten und dabei den eigenen Bedarf zu decken ohne die Produktion zu beeinträchtigen. Es stellt sich die Frage, ob alternative Ressourcen für die eigene Produktion in Frage kommen, um Kosten zu sparen. Im Folgenden geben wir die formale Definition für Congestion Spiele an, die erstmals 1973 von Rosenthal [9] eingeführt wurde.

Definition 2.2 (Congestion Spiel). *Ein Congestion Spiel Γ ist ein Tupel $(\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$, wobei $\mathcal{N} = \{1, \dots, n\}$ die Menge der Spieler beschreibt, \mathcal{R} mit $|\mathcal{R}| = m$ die Menge der Ressourcen, $(\Sigma_i)_{i \in \mathcal{N}} \subseteq 2^{\mathcal{R}}$ den Strategieraum eines Spielers $i \in \mathcal{N}$, und $d_r : \mathbb{N} \rightarrow \mathbb{N}$ die Kostenfunktion (delay) einer Ressource r .*

Für einen Zustand S definieren wir die Auslastung (congestion) $n_r(S)$ einer Ressource r durch $n_r(S) = |\{i | r \in S_i\}|$. Die Auslastung beschreibt die Anzahl der Spieler, die sich im Zustand S die Ressource r teilen. Wir gehen davon aus, dass das Ziel jedes Spielers ist, seine Gesamtkosten im Spiel zu minimieren und er dadurch in jedem Schritt egoistisch handelt, d.h. die für ihn beste Strategie wählt. Die Kosten $\delta_i(S)$ eines Spielers i in einem Zustand S sind gegeben durch $\delta_i(S) = \sum_{r \in S_i} d_r(n_r(S))$. Anschaulich bedeutet dies, dass sich ein Spieler i eine Menge von Ressourcen $\mathcal{R}' \subseteq \mathcal{R}$ wählt. Seine Kosten $\delta_i(S)$ summieren sich aus den Kosten der gewählten Ressourcen $r' \in \mathcal{R}'$. Die Kosten einer Ressource r hängen von der Auslastung $n_r(S)$ ab, welche die Anzahl der Spieler beschreibt, die r belegen.

Auf unser Eingangsbeispiel angewendet, können wir das Produktionsspiel $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$ formal definieren mit

- $\mathcal{N} = \{\text{Firma 1, Firma 2, Firma 3}\}$, für eine Konkurrenzsituation zwischen drei Firmen,
- $\mathcal{R} = \{\text{Eisen, Stahl, Gold, Silber}\}$ beschreibt die Menge der Ressourcen,
- $(\Sigma_i)_{i \in \mathcal{N}}$ beschreibt welche Firma, welche Rohstoffe einkaufen kann und
- $d_r : \mathbb{N} \rightarrow \mathbb{N}$ mit $d_{\text{Eisen}}(x) = x$, $d_{\text{Stahl}}(x) = 2x$, $d_{\text{Gold}}(x) = x^3$ und $d_{\text{Silber}}(x) = 2x^2$, wobei x die Nachfrage nach der Ressource r beschreibt, die sich in der Anzahl der Firmen ausdrückt, die r nachfragen. In diesem Beispiel verursachen die Ressourcen Gold und Silber höhere Kosten als Eisen.

2.2 Das Nash-Gleichgewicht

Ein weiterer wichtiger Begriff für diese Arbeit ist das *Nash-Gleichgewicht*. Das Nash-Gleichgewicht ist ein Konzept, das in nicht-kooperativen Spielen auftritt. Die Definition geht auf den Mathematiker John Forbes Nash zurück. Anschaulich beschreibt das Nash-Gleichgewicht einen Zustand eines Spiels, in dem kein Spieler einen Anreiz hat, seine Strategie zu ändern. Das heißt, dass eine Änderung seiner Strategie sich nicht positiv auf den Spieler auswirkt, sei es in Form der Erhöhung des eigenen Gewinns oder in Form eines Verlustes

für einen anderen Spieler. Aus dieser Beschreibung folgt, dass in einem Nash-Gleichgewicht jeder Spieler eine *beste Antwort* auf die Strategien der anderen Spieler spielt.

Definition 2.3 (beste Antwort). *Eine Strategie $S_i^* \in \Sigma_i$ ist eine beste Antwort für Spieler i auf den Zustand S , wenn für alle Strategien $S'_i \neq S_i^* \in \Sigma_i$ gilt*

$$U_i(S \oplus S_i^*) \geq U_i(S \oplus S'_i).$$

Eine Sequenz bester Antworten ist eine Folge von besten Antworten, die in jedem Schritt den Gewinn des Spielers i erhöhen.

In Congestion Spielen ist es unser Ziel, die entsprechende Kostenfunktion zu verringern. Es gilt analog

$$S_i^* \in \Sigma_i \text{ ist beste Antwort auf } S : \Leftrightarrow \forall S'_i \in \Sigma_i : \delta_i(S \oplus S_i^*) \leq \delta_i(S \oplus S'_i).$$

Eine Sequenz bester Antworten reduziert die Kosten von Spieler i in jedem Schritt.

Mit Hilfe der Definition der besten Antwort können wir nun auch das Nash-Gleichgewicht formalisieren.

Definition 2.4 (Nash-Gleichgewicht). *Ein Zustand S heißt Nash-Gleichgewicht, falls*

$$\forall i \in \mathcal{N} : \forall S'_i \in \Sigma_i : U_i(S) \geq U_i(S \oplus S'_i).$$

In einem Congestion Spiel Γ nennen wir einen Zustand S ein Nash-Gleichgewicht, falls

$$\forall i \in \mathcal{N} : \forall S'_i \in \Sigma_i : \delta_i(S) \leq \delta_i(S \oplus S'_i).$$

Zuvor haben wir in der Definition des Spieles reine und gemischte Strategien unterschieden. Diese Differenzierung findet sich auch für Nash-Gleichgewichte wieder. Für gemischte Strategien ändert sich Definition 2.4 nur in der Betrachtung des Strategievektors. Anstelle einer expliziten Strategiesequenz betrachten wir eine Wahrscheinlichkeitsverteilung $p_i = (p_{i,1}, \dots, p_{i,t})$ mit $t = |\Sigma_i|$ auf dem Strategieraum Σ_i für einen Spieler i . Wir beschreiben mit $p_{i,j}$ die Wahrscheinlichkeit, dass die Strategie $S_{i,j} \in \Sigma_i$ gewählt wird. Eine gemischte Strategie $S_i = (p_1, \dots, p_n)$ ist genau dann ein Nash-Gleichgewicht, wenn kein Strategiewechsel für einen Spieler i sich positiv für ihn auswirkt. Während einige Spiele kein reines Nash-Gleichgewicht besitzen, besitzt jedes Spiel ein Nash-Gleichgewicht mit gemischten Strategien [12].

Die Erläuterung der gemischten Strategien sei nur aus Gründen der Vollständigkeit genannt. Wir werden uns im Folgenden auf die Betrachtung reiner Nash-Gleichgewichte beschränken und ihre Existenz in Congestion Spielen untersuchen. Daher implizieren wir fortan reine Nash-Gleichgewichte, wenn wir

von Nash-Gleichgewichten sprechen. Weiterhin werden wir analysieren, in welcher Komplexitätsklasse die Berechnung eines Nash-Gleichgewichtes in Congestion Spielen liegt. Hierbei werden die Komplexitätsklasse \mathcal{PLS} [6] und *Rosenthals Potentialfunktion* [9] von zentraler Bedeutung sein.

2.3 Die Klasse \mathcal{PLS}

Für die Analyse der Berechnung eines Nash-Gleichgewichtes in Netzwerk Congestion Spielen, benötigen wir die Definition der Komplexitätsklasse *Polynomial Local Search* (\mathcal{PLS}), die von Johnson, Papadimitriou und Yannakakis [6] eingeführt wurde. Sie fasst *lokale Suchprobleme* zusammen, deren Nachbarschaften in polynomieller Zeit auf bessere Lösungen durchsucht werden können. Zunächst beschreiben wir ein lokales Suchproblem formal.

Definition 2.5 (lokales Suchproblem). *Ein lokales Suchproblem Π ist gegeben durch eine Menge von Instanzen \mathcal{I}_Π . Für jede Instanz $I \in \mathcal{I}_\Pi$ definieren wir eine Menge von Lösungen $\mathcal{F}(I)$, eine Zielfunktion $c : \mathcal{F}(I) \rightarrow \mathbb{N}$ und für jede Lösung $S \in \mathcal{F}(I)$ eine Nachbarschaft $\mathcal{N}(S, I) \subseteq \mathcal{F}(I)$.*

Sei I eine Instanz eines lokalen Suchproblems. Dann suchen wir nach einer *lokal optimalen Lösung* S^* .

Definition 2.6 (lokal optimale Lösung). *Eine Lösung S^* ist genau dann lokal optimal, wenn es in ihrer Nachbarschaft $\mathcal{N}(S^*, I)$ keine strikt bessere Lösung gibt. Für ein Maximierungsproblem ist eine Lösung S' genau dann strikt besser als eine Lösung S , falls $c(S') > c(S)$ gilt. Für ein Minimierungsproblem ist S' strikt besser als S , falls $c(S') < c(S)$ gilt.*

Eine formale Beschreibung der Klasse \mathcal{PLS} bietet die folgende Definition.

Definition 2.7 (\mathcal{PLS}). *Ein lokales Suchproblem Π gehört zu der Klasse \mathcal{PLS} , wenn es für die folgenden Aufgaben Algorithmen gibt, die in polynomieller Zeit ausführbar sind.*

1. *Ein Algorithmus A berechnet für alle Instanzen $I \in \Pi$ eine initiale Lösung $S^0 \in \mathcal{F}(I)$,*
2. *Ein Algorithmus B berechnet für alle Instanzen $I \in \Pi$ und für alle Lösungen $S \in \mathcal{F}(I)$ den Wert $c(S)$ der Zielfunktion c .*
3. *Ein Algorithmus C bestimmt für alle Instanzen $I \in \Pi$ und für jede Lösung $S \in \mathcal{F}(I)$, ob S lokal optimal ist. Falls dies nicht der Fall ist, findet C eine bessere Lösung in $\mathcal{N}(S, I)$.*

Für die Definition der \mathcal{PLS} -Reduzierbarkeit führen wir einen *Übergangsgraphen* $TG(I)$ eines lokalen Suchproblems $I \in \Pi$ ein. $TG(I)$ enthält für jede Lösung $S \in \mathcal{F}(I)$ einen Knoten $v(S)$. Zwei Knoten $v(S^1)$ und $v(S^2)$ sind genau dann adjazent, wenn $S^2 \in \mathcal{N}(S^1, I)$ und $c(S^2) > c(S^1)$ gilt.

Definition 2.8 (\mathcal{PLS} -Reduktion). *Ein Problem $\Pi_1 \in \mathcal{PLS}$ ist \mathcal{PLS} -reduzierbar auf ein Problem $\Pi_2 \in \mathcal{PLS}$, wenn es polynomiell berechenbare Funktionen f und g gibt, sodass:*

1. f bildet Instanzen $I \in \Pi_1$ auf Instanzen $f(I) \in \Pi_2$ ab,
2. g bildet Tupel (S^2, I) mit $S^2 \in \mathcal{F}(f(I))$ auf Lösungen S^1 von I ab,
3. für alle Instanzen $I \in \Pi_1$ und alle Lösungen S^2 von $f(I)$ gilt, wenn S^2 ein lokales Optimum der Instanz $f(I)$ ist, dann ist $g(S^2, I)$ ein lokales Optimum zu I .

Definition 2.9 (\mathcal{PLS} -vollständig). *Ein lokales Suchproblem $\Pi \in \mathcal{PLS}$ ist \mathcal{PLS} -vollständig, wenn alle Probleme in \mathcal{PLS} \mathcal{PLS} -reduzierbar auf Π sind.*

Bisher ist für kein \mathcal{PLS} -vollständiges Problem P ein Algorithmus bekannt, der ein lokales Optimum für P in polynomieller Zeit berechnet. Weiterhin haben Johnson und Papadimitriou [6] gezeigt, dass $\mathcal{NP} = \text{co-}\mathcal{NP}$ gilt, falls ein Problem in \mathcal{PLS} auch \mathcal{NP} -schwer ist.

Kapitel 3

Congestion Spiele und das Nash-Gleichgewicht

3.1 Rosenthals Potentialfunktion

R. W. Rosenthal [9] hat bewiesen, dass jedes Congestion Spiel ein Nash-Gleichgewicht besitzt und dieses in endlich vielen Schritten erreicht werden kann. Dies bedeutet, dass es nur endlich viele Verbesserungsschritte für einen Spieler gibt. Diese Erkenntnis basiert auf einem Beweis, der *Rosenthals Potentialfunktion* verwendet.

Theorem 3.1. *Jedes Congestion Spiel besitzt mindestens ein Nash-Gleichgewicht [9].*

Beweis. Der Beweis von Theorem 3.1 verwendet die Methodik der Potentialfunktion. Seien S ein Zustand eines Congestion Spiels und $n_r(S)$ die Auslastung der Ressource r in einem Zustand S . $n_r(S)$ beschreibt die Anzahl der Spieler, die die Ressource r verwenden. Für jeden Zustand S definiere das Potential $\Phi : \Sigma_1 \times \cdots \times \Sigma_n \rightarrow \mathbb{N}_0$ mit

$$\Phi(S) = \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i).$$

$\Phi(S)$ wird auch Rosenthals Potentialfunktion genannt. Zum Beweis des Theorems 3.1 beweisen wir zunächst folgendes Lemma.

Lemma 3.2. *Sei S ein beliebiger Zustand in einem Congestion Spiel Γ . Sei nun $S \oplus S'_i$ der Zustand zu dem wir von Zustand S übergehen und in dem Spieler i seine Kosten δ_i um $\Delta \geq 0$ verringert. Dann gilt $\Phi(S \oplus S'_i) = \Phi(S) - \Delta$.*

Beweis. Rosenthals Potentialfunktion stellt eine Summierung über die Ressourcen $r \in \mathcal{R}$ da. Für jede Ressource r werden die Kosten $d_r(i)$ für jede

Anzahl $1 \leq i \leq n_r(S)$ aufsummiert. Dies bedeutet, dass die Kosten einer Resource für jede Anzahl von Belegungen bis zur tatsächlichen Belegung $n_r(S)$ in Zustand S einbezogen werden.

Wir betrachten nun den Fall, dass wir von Zustand S auf den Zustand $S' = S \oplus S'_i$ übergehen. Dann gilt nach Voraussetzung für Spieler i

$$\delta_i(S') = \delta_i(S) - \Delta, \quad (3.1)$$

da Spieler i seine Kosten durch einen Strategiewechsel in S' um $\Delta \geq 0$ senken konnte. Die Kosten des Spielers i werden beschrieben durch

$$\delta_i(S) = \sum_{r \in S_i} d_r(n_r(S)).$$

Betrachte den Summanden $\sum_{r \in \mathcal{R}} d_r(n_r(S))$ der Potentialfunktion $\Phi(S)$. Die Kostenfunktion $\delta_i(S)$ des Spielers i lässt sich hierin wiederfinden:

$$\sum_{r \in \mathcal{R}} d_r(n_r(S)) = \sum_{r' \in \mathcal{R} \setminus S_i} d_{r'}(n_{r'}(S)) + \sum_{r \in S_i} d_r(n_r(S)). \quad (3.2)$$

Ohne Beschränkung der Allgemeinheit können wir annehmen, dass sich die Strategien der anderen Spieler $j \neq i$ bei dem Übergang auf Zustand S' nicht ändern und somit ihre Kosten δ_j gleich bleiben. Es gilt

$$\begin{aligned} \Phi(S') &= \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S')} d_r(i) \\ &= \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S')-1} d_r(i) + \sum_{r \in \mathcal{R}} d_r(n_r(S')) \\ &\stackrel{(3.2)}{=} \underbrace{\sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S')-1} d_r(i) + \sum_{r' \in \mathcal{R} \setminus S_i} d_{r'}(n_{r'}(S'))}_{\mathcal{K}} + \sum_{r \in S_i} d_r(n_r(S')) \\ &= \mathcal{K} + \delta_i(S') \\ &\stackrel{(3.1)}{=} \underbrace{\mathcal{K} + \delta_i(S)}_{\Phi(S)} - \Delta \\ &= \Phi(S) - \Delta. \end{aligned}$$

□

Lemma 3.2 zeigt, dass Rosenthals Potentialfunktion $\Phi(S)$ sich in einem Verbesserungsschritt genau um die Differenz verringert, um die sich $\delta_i(S)$ verringert.

Aus der Definition von δ_i folgt, dass $\delta_i(S)$ nur ganzzahlige Werte annehmen kann. Hieraus folgt, dass nur endlich viele Verbesserungsschritte durchgeführt werden können und daher ein Nash-Gleichgewicht nach maximal $\Phi(S)$ Schritten erreicht wird. \square

Aus Theorem 3.1 folgt, dass ein Nash-Gleichgewicht für Congestion Spiele berechnet werden kann, indem man Rosenthals Potentialfunktion minimiert. Fabrikant, Papadimitriou und Talwar [3] haben gezeigt, dass die Berechnung eines Nash-Gleichgewichtes für Congestion Spiele im Allgemeinen \mathcal{PLS} -vollständig ist. Wir erweitern diese Erkenntnis und zeigen, dass die Optimierung von Rosenthals Potentialfunktion *stark \mathcal{NP} -schwer* ist.

Definition 3.3 (stark \mathcal{NP} -schwer). *Sei Π ein Berechenbarkeitsproblem. Dann ist Π stark \mathcal{NP} -schwer, wenn Π bereits \mathcal{NP} -schwer für Instanzen mit Koeffizienten $c_i \in \mathbb{Z}$ ist, für die gilt, dass $C := \max_i |c_i|$ polynomiell in der Eingabelänge liegt.*

Ein Beispiel für ein stark \mathcal{NP} -schweres Problem ist das bekannte *Travelling Salesperson*-Problem, da es schon für geringe Kantengewichte stark \mathcal{NP} -schwer ist [8]. Um zu zeigen, dass die Minimierung von Rosenthals Potentialfunktion für Congestion Spiele stark \mathcal{NP} -schwer ist, reduzieren wir das *Not-All-Equal-3SAT* Problem auf das Minimierungsproblem der Potentialfunktion. Das *Not-All-Equal-3SAT* Problem ist eine abgewandelte Version des *3SAT* Problems.

Definition 3.4 (NAE-3SAT). *Das Not-All-Equal-3SAT Problem erhält als Eingabe einen logischen Ausdruck $\varphi = \bigwedge_{i=1}^n \bigvee_{j=1}^3 (\neg)x_{ij}$ in konjunktiver Normalform, bestehend aus n Klauseln. Jede Klausel C_i , $1 \leq i \leq n$, besteht aus genau drei Literalen. Das Ziel ist es, eine Belegung x^* für φ zu finden, sodass φ erfüllt ist und in jeder Klausel C_i nicht alle Literale die gleiche Belegung besitzen.*

Eine Instanz für NAE-3SAT ist zum Beispiel der Ausdruck

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4).$$

Die Belegung $x = (x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$ erfüllt den Ausdruck φ , jedoch besitzen drei Literale in der Klausel $(x_1 \vee x_2 \vee x_3)$ die gleiche Belegung. Um daher **true** für das NAE-3SAT liefern zu können, muss es eine Belegung geben, die in keiner Klausel zulässt, dass drei Literale den gleichen Wert besitzen. Für unser Beispiel ist $x^* = (1, 1, 0, 0)$ eine gültige Lösung. Somit ist φ für das NAE-3SAT erfüllbar.

Es ist bekannt, dass NAE-3SAT stark \mathcal{NP} -schwer ist [10]. Damit können wir folgendes Theorem beweisen.

Theorem 3.5. *Es ist im Allgemeinen stark \mathcal{NP} -schwer Rosenthals Potentialfunktion für Congestion Spiele zu minimieren.*

Beweis. Im Folgenden bezeichnen wir das Problem, Rosenthals Potentialfunktion zu minimieren, einfachheitshalber als MINROS. Um zu zeigen, dass MINROS für beliebige Congestion Spiele stark \mathcal{NP} -schwer ist, führen wir eine Reduktion von NAE-3SAT auf MINROS durch. Sei φ eine Instanz des NAE-3SAT Problems mit m Klauseln. Wir führen den Wert des Gewichtes $p(\varphi)$ ein, der die Anzahl erfüllter Klauseln für die Instanz φ , im Sinne des NAE-3SAT, beschreibt. Wir wollen zeigen, dass eine Belegung φ genau dann erfüllt, wenn Rosenthals Potentialfunktion Φ minimiert wird. Dies entspricht dem Zustand, dass $p(\varphi)$ maximal ist.

Wir bilden die NAE-3SAT Instanz φ wie folgt auf ein Congestion Spiel Γ ab:

- Jedes Literal x_i aus φ wird auf einen Spieler i in Γ abgebildet.
- In jeder Klausel C_j , $1 \leq j \leq m$ können die enthaltenen Literale den Wert 0 oder 1 annehmen. Wir übertragen die beiden Werte auf Ressourcen $r_{C_j}^0$ und $r_{C_j}^1$ für jede Klausel C_j in Γ .
- Daraus ergibt sich, dass jeder Spieler i die Strategien $S_i^0 = \{r_{C_j}^0 | x_i \in C_j\}$ und $S_i^1 = \{r_{C_j}^1 | x_i \in C_j\}$ besitzt.
- Ist eine Klausel C_j durch eine Belegung seiner enthaltenen Literale erfüllbar, in der nicht alle Literale den Wert 0 oder alle Literale den Wert 1 besitzen, so wird dem Wert $p(\varphi)$ das Gewicht 1 hinzugefügt. Für Γ konstruieren wir die Kostenfunktion $d_{r_{C_j}^a}$ für eine Ressource $r_{C_j}^a$ mit $a \in \{0, 1\}$ durch

$$d_{r_{C_j}^a}(k) = \begin{cases} 1 & \text{falls } k = 3, \\ 0 & \text{sonst,} \end{cases} \quad (3.3)$$

wobei k die Anzahl der Spieler beschreibt, die $r_{C_j}^a$ belegen.

Hieraus ergibt sich ein Congestion Spiel $\Gamma = (\mathcal{N}, \mathcal{R}, \Sigma_i, d_{r_{C_j}^a})$ mit

- $\mathcal{N} = \{1, \dots, n\}$,
- $\mathcal{R} = \{r_{C_j}^0, r_{C_j}^1 | 1 \leq j \leq m\}$,
- $\Sigma_i = \{S_i^0, S_i^1\}_{i \in \mathcal{N}}$,
- $d_{r_{C_j}^a}$ mit $a \in \{0, 1\}$.

Die Reduktion ist in polynomieller Zeit durchführbar, da die Komponenten und Werte aus NAE-3SAT genau auf eine individuelle Komponente in einem Congestion Spiel Γ direkt abgebildet werden.

Es bleibt zu zeigen, dass der Wert $p(\varphi)$ für eine NAE-3SAT Instanz φ genau dann maximal ist, wenn Rosenthals Potentialfunktion Φ minimiert wird. Sei hierfür φ eine Instanz von NAE-3SAT mit Wert $p(\varphi)$. Das bedeutet, dass $p(\varphi)$ Klauseln erfüllt sind, in denen nicht alle Literale in einer erfüllten Klausel den selben Wert haben. Daraus folgt, dass genau $m - p(\varphi)$ viele Klauseln im Sinne von NAE-3SAT nicht erfüllt sind. Wir nennen die Menge, die genau diese Klauseln enthält, \bar{C} . Daraus folgt, dass für alle Klauseln $\bar{C}_j \in \bar{C}$, $1 \leq j \leq m - p(\varphi)$, eine Ressource $r_{\bar{C}_j}^a$ in einer Klausel C_j von drei Spielern belegt wird. Dadurch tragen die Klauseln $\bar{C}_j \in \bar{C}$ den Wert 1 zu Rosenthals Potentialfunktion bei. Mit dieser Erkenntnis und der obigen Kostenfunktion 3.3 für Γ folgt für Rosenthals Potentialfunktion

$$\begin{aligned}
\Phi(S) &= \sum_{j=1}^m \sum_{a \in \{0,1\}} \sum_{k=1}^{n_{r_{\bar{C}_j}^a}(S)} d_{r_{\bar{C}_j}^a}(k) \\
&= \sum_{j=1}^m \left(\sum_{k=1}^{n_{r_{\bar{C}_j}^0}(S)} d_{r_{\bar{C}_j}^0}(k) + \sum_{k=1}^{n_{r_{\bar{C}_j}^1}(S)} d_{r_{\bar{C}_j}^1}(k) \right) \\
&= \sum_{\bar{C}_j \in \bar{C}} \left(\sum_{k=1}^{n_{r_{\bar{C}_j}^0}(S)} d_{r_{\bar{C}_j}^0}(k) + \sum_{k=1}^{n_{r_{\bar{C}_j}^1}(S)} d_{r_{\bar{C}_j}^1}(k) \right) \\
&= \sum_{\bar{C}_j \in \bar{C}} 1 \\
&= |\bar{C}|.
\end{aligned}$$

Das Potential ist also nur von den Klauseln in φ abhängig, für die alle Literale den gleichen Wert annehmen.

Sei nun $\bar{C}_k \in \bar{C}$ eine Klausel, die erfüllt ist, aber für alle Literale die gleiche Belegung besitzt. Invertieren wir den Wert eines Literals in \bar{C}_k , so ist \bar{C}_k im Sinne von NAE-3SAT erfüllt und fügt dem Gewicht $p(\varphi)$ den Wert 1 hinzu. Gleichzeitig bedeutet dies für Γ , dass es eine Klausel weniger gibt, in denen drei Spieler dieselbe Ressource belegen. Daraus folgt, dass das Potential Φ um den Wert 1 sinkt. Hieraus folgt, dass eine Änderung in der Belegung von φ äquivalente Auswirkungen auf den Wert von Φ hat. Damit ist gezeigt, dass Rosenthals Potentialfunktion genau dann minimiert wird, wenn φ in NAE-3SAT erfüllbar ist. \square

Diese Erkenntnis impliziert direkt das folgende Korollar.

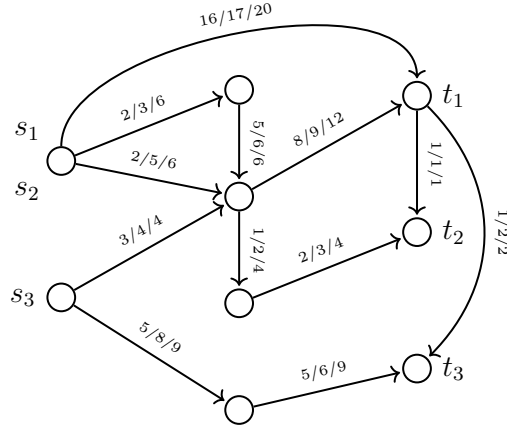
Korollar 3.6. *Für Congestion Spiele ist es im Allgemeinen stark \mathcal{NP} -schwer ein Nash-Gleichgewicht mittels Minimierung der Rosenthal Potentialfunktion zu finden.*

3.2 Netzwerk Congestion Spiele

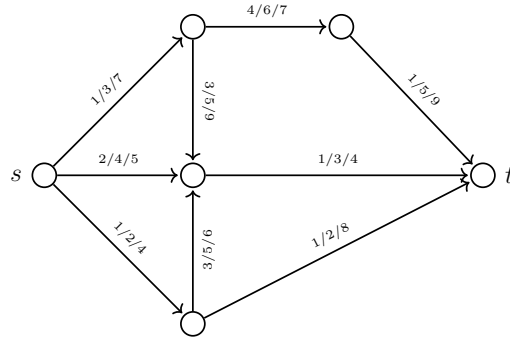
In diesem Kapitel betrachten wir eine Form von Congestion Spielen genauer. Die Ressourcen bestehen aus Wegen oder Pfaden. Diese Art der Ressourcenverteilung findet auch in der Wirtschaft ihre Anwendung. Wir betrachten die sogenannten *Netzwerk Congestion Spiele*.

Ein Beispiel für ein Netzwerk Congestion Spiel ist die Routenplanung. Wir haben ein Urlaubsziel und möchten dies mit dem Auto möglichst schnell und kostensparend erreichen. Es liegt nahe, dass wir uns die schnellste Verbindung über beispielsweise eine Autobahn aussuchen. Jedoch kann es sein, dass diese viel befahren ist und uns dadurch nicht nur hohe Zeitkosten, sondern auch Kosten für mehr Sprit oder Mautkosten anfallen. Hier wäre es sinnvoll zu betrachten, ob andere Wege, die weniger befahren sind, uns schneller ans Ziel bringen. Dieses Problem kennen auch Speditionen und Logistikunternehmen, die Güter schnell liefern müssen, sodass möglichst keine Produktionsverzögerungen auftreten. Die Entscheidung, welche Route von A nach B genommen wird, hängt nicht nur davon ab wie lang diese ist, sondern auch davon, wie viele andere Akteure die Route oder einen Teil der Route beanspruchen. Dies ist auch interessant, wenn wir den Fall betrachten, in dem alle Akteure den selben Start- und Zielort haben. Beispielsweise ist dies bei öffentlichen Großveranstaltungen wie Konzerten, Public Viewings oder Fußballspielen oft der Fall.

Ein Netzwerk Congestion Spiel besteht aus einem gerichteten Graphen $G = (V, E)$ und n Spielern. Jedem Spieler $i \in \{1, \dots, n\}$ sind zwei Knoten $s, t \in V$ zugeordnet, eine Quelle und eine Senke. Ziel jedes Spielers ist es, einen Weg von s nach t in G zu finden, der möglichst geringe Kosten mit sich bringt. Die Kosten einer Kante $e \in E$ sind davon abhängig, wie viele Spieler e nutzen. Die Kosten werden durch eine Funktion $d_e : \mathbb{N} \rightarrow \mathbb{N}$ beschrieben. Der Strategieraum Σ_i eines Spielers i besteht demnach aus allen möglichen s - t -Wegen. Wir unterscheiden zwischen *asymmetrischen* und *symmetrischen* Netzwerk Congestion Spielen. In einem asymmetrischen Netzwerk Congestion Spiel gibt es mehrere Quell- und Senkknoten und die Spieler verfolgen unter Umständen unterschiedliche Pfade, die jedoch gleiche Kanten benutzen können. In symmetrischen Netzwerk Congestion Spielen gibt es genau einen Quellknoten und einen Senkknoten für alle Spieler. Hieraus folgt, dass alle Spieler den gleichen Strategieraum besitzen. In Abbildung 3.1 sind jeweils ein asymmetrisches und ein symmetrisches Netzwerk Congestion Spiel dargestellt.



(a) Ein asymmetrisches Netzwerk Congestion Spiel mit drei Spielern. Spieler 1 und 2 teilen sich einen Startknoten. Jeder Spieler besitzt einen individuellen Endknoten.



(b) Ein symmetrisches Netzwerk Congestion Spiel mit drei Spielern, einem Quellknoten s und einer Senke t .

Abbildung 3.1: Beispiele für Netzwerk Congestion Spiele: Die Kanten der Spiele sind mit den Kosten der Kanten bei einer Auslastung von $1/2/3$ Spielern beschriftet.

Wir werden in den folgenden Abschnitten zeigen, dass es im Allgemeinen \mathcal{PLS} -vollständig ist, ein Nash-Gleichgewicht für Congestion Spiele mit monoton steigenden Kostenfunktionen zu berechnen. Dem gegenüber stellt sich das Resultat von Fabrikant, Papadimitriou und Talwar [3], dass ein Nash-Gleichgewicht für symmetrische Netzwerk Congestion Spiele in polynomieller Zeit berechenbar ist. Mit Theorem 3.5 haben wir gezeigt, dass es im Allgemeinen \mathcal{NP} -schwer ist ein Nash-Gleichgewicht mittels Rosenthals Potentialfunktion zu finden. Dieses Ergebnis weiten wir auf asymmetrische Netzwerk Congestion Spiele aus.

3.2.1 Allgemeine Netzwerk Congestion Spiele

Die Frage nach der Komplexität der Berechnung eines Nash-Gleichgewichtes in Netzwerk Congestion Spielen ist gut untersucht. Nimmt man an, dass die Kostenfunktionen der Netzwerk Congestion Spiele monoton steigend sind, ergibt sich, dass es \mathcal{PLS} -vollständig ist, ein Nash-Gleichgewicht für Netzwerk Congestion Spiele zu berechnen. Wir stellen hierzu die Beweise von Ackermann, Röglin und Vöcking [1] vor.

Theorem 3.7. *Die Berechnung eines Nash-Gleichgewichtes für Netzwerk Congestion Spiele mit monoton steigenden linearen Kostenfunktionen ist \mathcal{PLS} -vollständig [1].*

Der Beweis zu Theorem 3.7 verwendet eine Reduktion auf *quadratische Schwellenwertspiele*. Schwellenwertspiele gehören zur Klasse der Congestion Spiele. Die Menge der Ressourcen \mathcal{R} eines Schwellenwertspiels wird in zwei disjunkte Teilmengen \mathcal{R}_{in} und \mathcal{R}_{out} eingeteilt. Die Menge \mathcal{R}_{out} enthält für jeden Spieler $i \in \mathcal{N}$ eine Ressource r_i . Jeder Ressource $r_i \in \mathcal{R}_{out}$ wird ein Kostenwert T_i zugewiesen, der den Schwellenwert für den Spieler i beschreibt. Jeder Spieler hat genau zwei Strategien $S_i^{out} = \{r_i\}$ und $S_i^{in} \subseteq \mathcal{R}_{in}$. Solange der Kostenwert der Strategie S_i^{in} kleiner als sein Schwellenwert T_i ist, bevorzugt Spieler i die Strategie S_i^{in} . *Quadratische Schwellenwertspiele* sind Schwellenwertspiele, in denen ein Spieler i gegenüber jedem Spieler $j \neq i$ eine eigene Strategie besitzt, sodass für jeden Spieler i gilt $S_i^{in} = \{r_{ij} | j \in \mathcal{N}, j \neq i\}$. Schäffer et al. [11] zeigen, dass die Berechnung eines Nash-Gleichgewichtes für quadratische Schwellenwertspiele mit monoton steigenden Kostenfunktionen \mathcal{PLS} -vollständig ist. Diesen Erkenntnis nutzen wir, um Theorem 3.7 zu beweisen.

Beweis zu Theorem 3.7. Um zu zeigen, dass es \mathcal{PLS} -vollständig ist ein Nash-Gleichgewicht in einem Netzwerk Congestion Spiel mit monoton steigender Kostenfunktion zu berechnen, verwenden wir eine \mathcal{PLS} -Reduktion von quadratischen Schwellenwertspielen auf Netzwerk Congestion Spiele.

Sei Γ ein quadratisches Schwellenwertspiel mit n Spielern. Wir bilden Γ auf ein Netzwerk Congestion Spiel Γ_N ab. Das Netzwerk zu Γ_N besteht aus dem unteren linken Dreieck eines $n \times n$ Gitters, wobei die Knoten auf der Hauptdiagonalen des Gitters einbezogen werden. Sei $1 \leq i, j \leq n$, dann bezeichne $v_{i,j}$ den Knoten in der i -ten Zeile und in der j -ten Spalte. Für jeden Spieler i in Γ führen wir einen Spieler i in Γ_N ein, der einen Startknoten $s_i = v_{i,1}$ und einen Endknoten $t_i = v_{n,i}$ besitzt. Die Kanten des Netzwerks werden durch die Menge E beschrieben mit

$$\begin{aligned} E = & \{(v_{i,j}, v_{i+1,j}) | 1 \leq i < n, 1 \leq j \leq n\} \\ & \cup \{(v_{i,j}, v_{i,j+1}) | 1 < i < n, 1 < j < n\} \\ & \cup \{(s_i, t_i) | \text{für alle Spieler } i\}. \end{aligned}$$

Die Kanten (s_i, t_i) werden auch als Schwellenwertkante für Spieler i bezeichnet. Durch die Konstruktion des Netzwerkes ist es lediglich Spieler i möglich die eigene Schwellenwertkante zu verwenden.

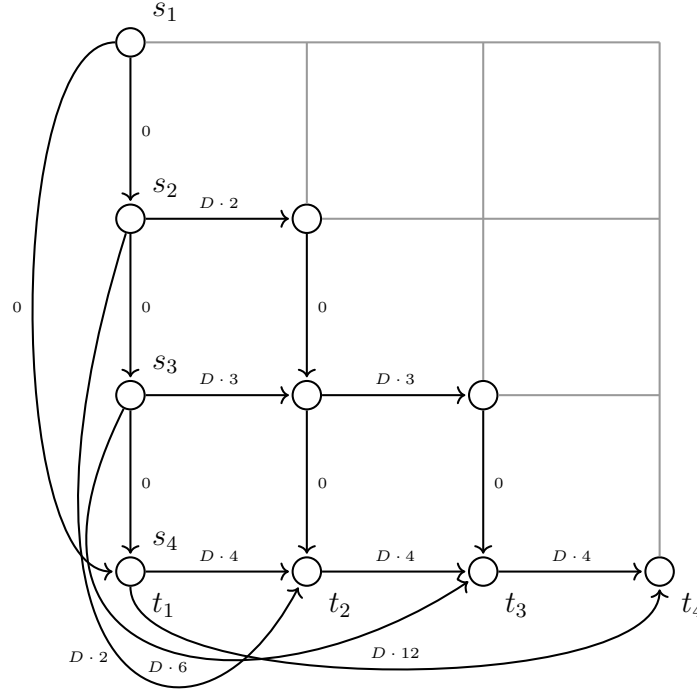


Abbildung 3.2: Das konstruierte Netzwerk zu Γ_N mit $n = 4$ angelehnt an [1]. Die Kanten sind mit den entsprechenden Kantengewichten beschriftet: Eine horizontale Kante in Zeile i erhält Kosten $D \cdot i$, eine vertikale Kante erhält Kosten 0 und eine Schwellenwertkante des Spielers i erhält Kosten $D \cdot i \cdot (i - 1)$.

Unser Ziel ist es nun, einem Spieler i nur zwei relevante Strategien zu ermöglichen. Das bedeutet, dass wir die Wege für Spieler i von s_i nach t_i auf zwei für i rentable Wege beschränken möchten. Einer dieser beiden Wege ist zunächst der direkte Weg von s_i nach t_i über die Schwellenwertkante (s_i, t_i) . Der zweite Weg ist der Zeilen-Spalten-Weg von s_i nach t_i . Dieser folgt der Methodik, zunächst von s_i aus horizontal den Kanten in Zeile i von Spalte 1 bis zu Spalte i zu folgen, und danach vertikal den Kanten in Spalte i bis Zeile n zu folgen, bis schließlich t_i erreicht wird. Wir erreichen die Beschränkung auf die beiden Wege, indem wir die Kostenwerte für alle anderen s_i - t_i -Pfade sehr hoch setzen, sodass es sich hinsichtlich keiner Strategie der Gegenspieler lohnt, diese Pfade zu wählen. Das heißt, dass für jeden Spieler i nur die Schwellenwertkante und der Zeilen-Spalten-Weg Strategien sind, die nicht dominiert werden.

Hierzu erhalten alle horizontal verlaufenden Kanten der Form $(v_{i,j}, v_{i,j+1})$ mit $1 < i < n, 1 < j < i$ die Kosten $D \cdot i$ für eine Zeile i und ein hinreichend großes $D \in \mathbb{R}$. Die vertikal verlaufenden Kanten der Form $(v_{i,j}, v_{i+1,j})$ mit $1 \leq i < n, 1 \leq j \leq n$ erhalten konstante Kosten in Höhe von 0. Die Kosten für die Schwellenwertkanten (s_i, t_i) erhalten konstante Kosten von $D \cdot i \cdot (i - 1)$. Der Zeilen-Spalten-Weg verursacht Kosten nur auf den horizontalen Kanten. Da er genau $i - 1$ horizontale Kanten passieren muss, entstehen für diese Strategie Kosten von $(D \cdot i) \cdot (i - 1)$, die genau den Kosten der Schwellenwertkante entsprechen. Eine graphische Darstellung des konstruierten Netzwerkes bietet Abbildung 3.2.

Im nächsten Schritt weisen wir auch den Knoten des Netzwerkes Kosten zu. Dies ist möglich, obwohl in einem Netzwerk Congestion Spiel im Allgemeinen nur Kanten Ressourcen darstellen. Dies ist erlaubt, da wir die Ressource eines Knotens durch eine Kante simulieren können. Dafür führen wir für jeden Knoten $v_{i,j}$ einen Hilfsknoten $h_{v_{i,j}}$ ein, sowie eine Kante $(v_{i,j}, h_{v_{i,j}})$. Wir verändern das Netzwerk nun so, dass alle ausgehenden Kanten aus $v_{i,j}$ ausgehende Kanten des Hilfsknotens $h_{v_{i,j}}$ werden. Die neue Kante $(v_{i,j}, h_{v_{i,j}})$ simuliert den Knoten $v_{i,j}$ als Ressource. Durch diesen Umbau erhalten wir ein neues Netzwerk, in dem alle Ressourcen aus Kanten bestehen. Die Kosten eines Knotens $v_{i,j}$ mit $1 \leq i < j \leq n$ in Γ_N entsprechen den Kosten der Ressource $r_{i,j} \in \mathcal{R}_{in}$ im quadratischen Schwellenwertspiel Γ . Abbildung 3.3 stellt dies graphisch dar.

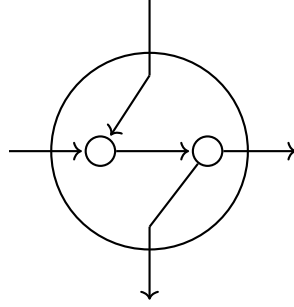


Abbildung 3.3: Die Ressource eines Knotens kann durch eine Kante simuliert werden, sodass alle Ressourcen des Netzwerkes aus Kanten bestehen [1].

Damit erreichen wir, dass der Zeilen-Spalten-Weg des Spielers i genau der Strategie S_i^{in} aus Γ entspricht. Erhöhen wir die Kosten der Schwellenwertkante (s_i, t_i) um die Kosten T_i der Ressource $r_i \in \mathcal{R}_{in}$ auf $D \cdot i \cdot (i - 1) + T_i$, dann entspricht die Schwellenwertkante genau der Strategie S_i^{out} .

Wählen wir D größer als die Summe aller Kosten in Γ , werden für jeden Spieler alle Strategien außer der Schwellenwertkante und dem Zeilen-Spalten-Weg dominiert, sodass sie nicht beachtet werden müssen. Dadurch bleiben lediglich zwei Strategien für jeden Spieler übrig. Wir haben damit einen Iso-

morphismus zwischen den Strategieräumen der Spieler in Γ_N zu den Strategieräumen in Γ konstruiert. Das heißt, dass die Strategien und Kostenfunktionen der Ressourcen der beiden Congestion Spiele bijektiv aufeinander abgebildet werden können. Insbesondere befindet sich Γ_N genau dann in einem Nash-Gleichgewicht, wenn Γ sich in einem Nash-Gleichgewicht befindet. Daraus folgt unsere Behauptung. \square

Der Beweis des vorausgegangenen Theorems 3.7 beschränkt sich auf gerichtete Graphen. Das folgende Theorem zeigt jedoch, dass das Ergebnis der \mathcal{PLS} -Vollständigkeit für Netzwerk Congestion Spiele mit linearen Kostenfunktionen auch auf Netzwerk Congestion Spiele mit ungerichteten Kanten zutrifft.

Theorem 3.8. *Die Berechnung eines Nash-Gleichgewichtes für Netzwerk Congestion Spiele mit ungerichteten Kanten und monoton steigenden linearen Kostenfunktionen ist \mathcal{PLS} -vollständig [1].*

Beweis. Wir reduzieren ein quadratisches Schwellenwertspiel auf ein Netzwerk Congestion Spiel mit ungerichteten Kanten und linearer Kostenfunktion und zeigen damit, dass die Berechnung eines Nash-Gleichgewichtes für diese Spiele \mathcal{PLS} -vollständig ist.

Sei Γ ein quadratisches Schwellenwertspiel mit n Spielern. Wir erzeugen ein Netzwerk Γ_N wie im Beweis zu Theorem 3.7, mit dem Unterschied, dass die gerichteten Kanten zu ungerichteten Kanten werden. Außerdem führen wir für jede Schwellenwertkante $\{s_i, t_i\}$ einen Knoten s'_i ein, und ersetzen die Kante $\{s_i, t_i\}$ durch zwei Kanten $e_i^s = \{s'_i, s_i\}$ und $e_i^t = \{s'_i, t_i\}$. Jeder Spieler i in Γ ist auch ein Spieler in Γ_N mit Startknoten s'_i und Endknoten t_i .

Ähnlich zur Reduktion im Beweis zu Theorem 3.7 ist es auch hier unser Ziel, jeden Spieler auf zwei relevante Strategien zu beschränken. Um zu erreichen, dass nur der Zeilen-Spalten-Weg und die Schwellenwertkante für jeden Spieler relevante Strategien werden, passen wir die Kostenfunktionen der Kanten an. Die Kosten eines Knotens $v_{i,j}$, die den Kosten der Ressource $r_{i,j} \in \mathcal{R}_{out}$ entsprechen, sowie die Kosten der horizontalen Kanten von $D \cdot i$ bleiben erhalten. Die Kosten der vertikalen Kanten werden von 0 auf $n^n \cdot D$ geändert und die Kosten der Kante $e_i^s = \{s_i, s'_i\}$ betragen $n^{6n} \cdot D$. Die Konstruktion des Netzwerks Γ_N wird in Abbildung 3.4 für $n = 4$ veranschaulicht.

Nun möchten wir folgende Aussage zeigen: Entfernt man die direkte Kante e_i^t zwischen s'_i und t_i für jeden Spieler i , so ist der Zeilen-Spalten-Weg die einzige beste Antwort Strategie, die s'_i und t_i verbindet.

Seien D_i^{max} die maximalen Kosten, die auf dem Zeilen-Spalten-Weg entstehen. Bei der Betrachtung schließen wir jedoch die Kosten der Kante e_i^s aus. Dann können wir D_i^{max} nach oben beschränken durch

$$D_i^{max} < (i - 1) \cdot i \cdot D + (n - i)n^n D + D.$$

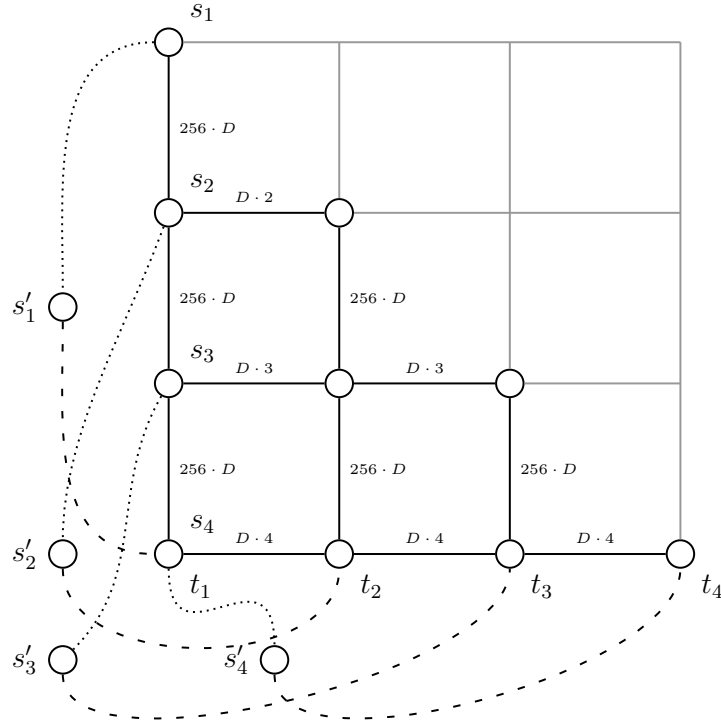


Abbildung 3.4: Das konstruierte Netzwerk zu Γ_N mit $n = 4$ angelehnt an [1]. Die Kanten sind mit den entsprechenden Kantengewichten beschriftet. Die Kanten $e_i^s = \{s_i, s'_i\}$ sind gepunktet dargestellt und tragen in diesem Beispiel Kosten von $4^{24} \cdot D$. Die Kanten $e_i^t = \{s'_i, t_i\}$ sind gestrichelt dargestellt und tragen Kosten $T_i + 4^{24}D + (4 - i)4^{24}D + (i - 1)iD$.

Die ersten beiden Summanden der Abschätzung entstehen durch die Traversierung der $(i - 1)$ horizontalen Kanten und der $n - i$ vertikalen Kanten auf dem Zeilen-Spalten-Weg. Wählt ein Spieler i einen anderen s'_i - t_i -Pfad als den Zeilen-Spalten-Pfad, so können folgende Fälle eintreffen:

1. Spieler i läuft über einen Knoten s'_j mit $j \neq i$.
2. Spieler i läuft über mehr als $n - i$ vertikale Kanten.
3. Spieler i läuft über eine horizontale Kante in einer Spalte $j > i$.

Tritt der erste Fall ein, so verursacht die Kante $\{s'_j, s_j\}$ Kosten von mindestens $n^{6n}D > D_i^{max}$. Im zweiten Fall betragen die Kosten mindestens $(n - i + 1)n^nD > D_i^{max}$. Tritt der dritte Fall ein, so nehmen wir an, dass Spieler i $n - i$ vertikale Kanten beansprucht, und mindestens eine Kante aus einer Spalte $j > i$. Dann ergeben sich Kosten in Höhe von

$$(n - i)n^nD + (i - 2)iD + jD \geq (n - i)n^nD + (i - 1)iD + D > D_i^{max}.$$

In allen drei Fällen übersteigen die Kosten D_i^{max} . Somit kann nur der Zeilen-Spalten-Weg eine beste Antwort sein. Wir wählen nun die Kosten der Kante e_i^t als

$$T_i + n^{6n}D + (n - i)n^nD + (i - 1)iD,$$

wobei T_i der Schwellenwert von Spieler i in Γ ist. Wir nehmen an, dass jeder Spieler beste Antworten spielt, das heißt, entweder die Kante e_i^t oder den Zeilen-Spalten-Weg wählt. Mit dieser Annahme gilt, dass sich die Kosten der Kante e_i^t genau aus der Summe des Schwellenwerts von Spieler i , den Kosten der Kante e_i^s und den Kosten des Zeilen-Spalten-Weges zusammen setzen. Damit entsprechen die Strategien in Γ_N genau den Strategien aus dem Schwellenwertspiel Γ . Aus dieser Reduktion folgt, dass ein Nash-Gleichgewicht in dem konstruierten Netzwerk Congestion Spiel äquivalent zu einem Nash-Gleichgewicht im quadratischen Schwellenwertspiel ist. \square

Aus den Theoremen 3.7 und 3.8 folgt, dass die Berechnung eines Nash-Gleichgewichtes für asymmetrische Netzwerk Congestion Spiele \mathcal{PLS} -vollständig ist.

Im Weiteren untersuchen wir die Komplexität das Nash-Gleichgewicht mittels Rosenthals Potentialfunktion zu berechnen. Wir beweisen, dass die Minimierung von Rosenthals Potentialfunktion für asymmetrische Netzwerk Congestion Spiele stark \mathcal{NP} -schwer ist.

Theorem 3.9. *Für asymmetrische Netzwerk Congestion Spiele ist es stark \mathcal{NP} -schwer ein Nash-Gleichgewicht mittels der Minimierung von Rosenthals Potentialfunktion zu finden.*

Beweis. Wir reduzieren das ARC-DISJOINT PATHS Problem, das stark \mathcal{NP} -schwer ist [4], auf ein asymmetrisches Netzwerk Congestion Spiel.

Das ARC-DISJOINT PATHS Problem erhält als Eingabe einen gerichteten Graphen $G = (V, E)$ und eine Menge von Knotenpaaren $(s_1, t_1), \dots, (s_n, t_n)$. Das Problem besteht darin zu entscheiden, ob es n kantendisjunkte Pfade P_1, \dots, P_n in G gibt, wobei P_i ein s_i - t_i -Pfad ist.

Sei nun eine Instanz des ARC-DISJOINT PATHS Problems mit $G = (V, E)$ und $(s_1, t_1), \dots, (s_n, t_n)$ gegeben. Wir reduzieren diese Instanz auf ein asymmetrisches Netzwerk Congestion Spiel Γ mit monoton steigenden Kostenfunktionen. Für jedes (s_i, t_i) -Paar führen wir einen Spieler i mit Startknoten s_i und Endknoten t_i in Γ ein. Die Ressourcen eines Spielers i bestehen aus allen Kanten $e \in E$, die von s_i aus erreichbar sind. Der Strategieraum Σ_i eines Spielers i besteht aus allen möglichen s_i - t_i -Pfaden. Weiterhin weisen wir jeder Ressource $r \in \mathbb{R}$ eine Kostenfunktion d_r zu mit

$$d_r(x) = \begin{cases} 0 & \text{falls } x \leq 1, \\ 1 & \text{sonst,} \end{cases} \quad (3.4)$$

wobei x die Anzahl der Spieler beschreibt, die r belegen. Die Reduktion erfolgt in polynomieller Zeit, da die Instanz des ARC-DISJOINT PATHS Problems eindeutig und direkt auf ein asymmetrisches Netzwerk Congestion Spiel abgebildet werden kann.

Es bleibt zu zeigen, dass Rosenthals Potentialfunktion genau dann minimiert wird, wenn das ARC-DISJOINT PATHS Problem **true** liefert. Existieren kantendisjunkte Pfade P_1, \dots, P_n für das ARC-DISJOINT PATHS Problem, so gilt für Rosenthals Potentialfunktion für das Netzwerk Congestion Spiel Γ

$$\Phi(S) = \sum_{r \in R} \sum_{i=1}^{n_r(S)} d_r(i) = 0, \quad (3.5)$$

da eine Kante $e \in E$ von maximal einem Spieler belegt wird. Somit erreicht Φ den minimalsten Wert und jeder Spieler i erhält minimale Kosten von 0.

Seien nun P_1, \dots, P_n keine kantendisjunkten Pfade für eine Instanz des ARC-DISJOINT PATHS Problems. Dann gilt $\Phi \geq 1$, da mindestens eine Ressource von mindestens zwei Spielern belegt wird. Sei P_i ein Pfad, der k Kanten besitzt, die jeweils von mindestens einem weiteren Pfad belegt werden. Falls es für P_i möglich ist, die Anzahl der geteilten Kanten k um Δ zu reduzieren, so sinkt auch Rosenthals Potentialfunktion Φ um Δ . Hieraus ergibt sich, dass Rosenthals Potentialfunktion genau dann minimiert wird, wenn eine Instanz des ARC-DISJOINT PATHS Problems möglichst wenige geteilte Kanten besitzt. Da bereits das Entscheidungsproblem, ob kantendisjunkte Pfade P_1, \dots, P_n existieren stark \mathcal{NP} -schwer ist, ist insbesondere eine Berechnung der P_1, \dots, P_n stark \mathcal{NP} -schwer. Es folgt unsere Behauptung. \square

Mit diesem Resultat haben wir gezeigt, dass sich das Ergebnis der \mathcal{NP} -Schwere der Minimierung von Rosenthals Potentialfunktion für allgemeine Congestion Spiele auf asymmetrische Netzwerk Congestion Spiele überträgt. Jedoch ist dies nicht der Fall für symmetrische Netzwerk Congestion Spiele, für die wir das Ergebnis von Fabrikant, Papadimitriou und Talwar [3] vorstellen werden.

3.2.2 Symmetrische Netzwerk Congestion Spiele

Nach der Betrachtung asymmetrischer Netzwerk Congestion Spiele, liegt es nahe, sich die Frage, ob Rosenthals Potentialfunktion effizient minimiert werden kann, auch für symmetrische Netzwerk Congestion Spiele zu stellen. In symmetrischen Netzwerk Congestion Spielen besitzen alle Spieler die gleichen Quell- und Senkknoten $s, t \in V$ und folglich den gleichen Strategieraum, der aus den s - t -Pfaden des Graphen G besteht.

Trotz der Erkenntnis aus Theorem 3.7, dass die Berechnung eines Nash-Gleichgewichtes für Netzwerk Congestion Spiele im Allgemeinen \mathcal{PLS} -vollständig ist, haben Fabrikant, Papadimitriou und Talwar [3] gezeigt, dass es für *symmetrische* Netzwerk Congestion Spiele einen Algorithmus gibt, der ein Nash-Gleichgewicht in polynomieller Zeit berechnet. Der Beweis beruht auf einer Reduktion eines symmetrischen Netzwerk Congestion Spiels auf das MIN-COST-FLOW Problem. Das MIN-COST-FLOW Problem besteht aus einem Graphen $G' = (V', E')$ mit einer Quelle $s \in V'$ und einer Senke $t \in V'$. Jede Kante $e = (u, v) \in E'$ hat eine Kapazität $c(u, v) > 0$ und einen Kostenwert $a(u, v)$ pro Flusseinheit. Mit $f(u, v)$ beschreiben wir, wie viele Flusseinheiten über eine Kante e fließen. Wir suchen einen Weg mit minimalen Kosten, der einen Fluss der Größe x von dem Quellknoten s zur Senke t transportiert. Unser Ziel ist es also die Gesamtkosten $\sum_{(u,v) \in E'} a(u, v) \cdot x$ zu minimieren. Dabei gelten folgende Nebenbedingungen:

- Die Anzahl an Flusseinheiten, die über eine Kante $e = (u, v)$ fließen, dürfen die Kapazität der Kante $c(u, v)$ nicht übersteigen: $f(u, v) \leq c(u, v)$.
- Eine Kante e kann keinen Fluss speichern: $\sum_{v \in V'} f(u, v) = 0$ für alle Knoten $u \neq s, t$. Daher muss der gesamte Fluss $f(u, v)$, der einfließt, auch wieder ausfließen: $f(u, v) = -f(u, v)$.
- Es kann unterwegs kein Fluss verloren gehen. Der Fluss, der von s aus transportiert wird, muss insgesamt an t ankommen. Es gilt $\sum_{v \in V'} f(s, v) = \sum_{u \in V'} f(u, t)$.

Es ist bekannt, dass es polynomielle Algorithmen für das MIN-COST-FLOW Problem gibt.

Theorem 3.10. *Es gibt einen Algorithmus, der ein Nash-Gleichgewicht für symmetrische Netzwerk Congestion Spiele in polynomieller Zeit berechnet [3].*

Beweis. Wir reduzieren das symmetrische Netzwerk Congestion Spiel auf das MIN-COST-FLOW Problem. Sei $\Gamma = (\mathcal{N} = \{1, \dots, n\}, N, (\Sigma_i)_{i \in \mathcal{N}}, d_e)$ ein symmetrisches Netzwerk Congestion Spiel mit dem Netzwerk $N = (V, E, s, t)$, wobei $s, t \in V$ die Quelle bzw. Senke beschreiben. Zu jeder Kante $e = (u, v)$ definieren wir eine Kostenfunktion d_e , die von der Anzahl der Spieler, die e verwenden, abhängig ist. Die Reduktion von Γ auf MIN-COST-FLOW geschieht wie folgt: Wir erweitern das Netzwerk N zu einem neuen Netzwerk N' . Für jede Kante $e = (u, v) \in E$ in dem Netzwerk N ersetzen wir e in N' durch n parallele Kanten e_1, \dots, e_n , die von u nach v führen. Eine Kante e_i besitzt eine Kapazität von 1 und trägt Kosten $d_e(i)$ mit $1 \leq i \leq n$. In Abbildung 3.5 wird diese Reduktion am Beispiel des symmetrischen Netzwerk Congestion Spiels aus Abbildung 3.1b dargestellt.

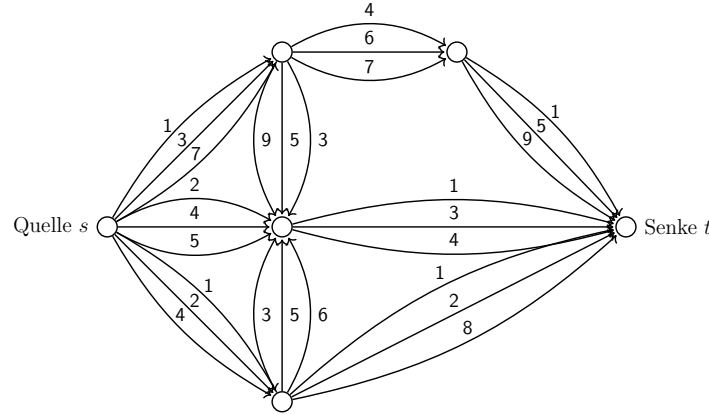


Abbildung 3.5: Bei der Reduktion des symmetrischen Netzwerk Congestion Spieles Γ auf das MIN-COST-FLOW Problem werden alle Kanten $e \in E$ durch jeweils n Kanten mit Kapazität 1 und Kosten $d_e(i)$, $0 \leq i \leq n$, ersetzt.

Wir betrachten einen ganzzahligen minimalen-Kosten-Fluss \mathcal{F} der Größe n . Da jede Kante eine Kapazität von 1 hat, muss der Fluss \mathcal{F} n Wege von s nach t verwenden, um einen Fluss der Größe n zu transportieren. Die n s - t -Wege entsprechen genau den Strategien der n Spieler in Γ . Da \mathcal{F} minimale Kosten hat, bevorzugt \mathcal{F} die Kanten mit den geringsten Kosten. Für zwei Kanten e_{i-1} und e_i im neuen Netzwerk N' gilt nach Konstruktion $d_e(i-1) \leq d_e(i)$. Daraus folgt, dass \mathcal{F} die günstigsten Kanten einer Kantenmenge e_1, \dots, e_n enthält, sodass e im neuen Netzwerk Kosten von $\sum_{i=1}^{n_e(S)} d_e(i)$ verursacht. Für den gesamten Fluss \mathcal{F} entstehen dadurch Kosten

$$\sum_{e \in E} \sum_{i=1}^{n_e(S)} d_e(i) = \Phi(S).$$

Da \mathcal{F} minimale Kosten hat und diese genau Rosenthals Potentialfunktion entsprechen, entspricht ein minimaler Fluss der Größe n einer Minimierung von $\Phi(S)$. Daraus folgt, dass wir ein Nash-Gleichgewicht gefunden haben, in dem die Kosten für jeden Spieler lokal optimal sind. \square

Es folgt direkt aus Theorem 3.10, dass es in polynomieller Zeit möglich ist, Rosenthals Potentialfunktion für symmetrische Netzwerk Congestion Spiele zu minimieren. Edmonds und Karp [2] haben 1972 einen Algorithmus entwickelt, der das MIN-COST-FLOW Problem in polynomieller Zeit löst, indem sukzessive kürzeste Wege gefunden werden. Der Algorithmus hat eine Laufzeit von $O(m^2n)$, wobei n die Anzahl der Knoten in dem Flussnetzwerk G und m die Anzahl der Kanten in G sind. Seitdem wurden zahlreiche Algorithmen entwickelt, um das MIN-COST-FLOW Problem zu lösen. Dazu gehören unter anderem Netzwerk Simplex Algorithmen [7].

Kapitel 4

Fazit

In dieser Arbeit haben wir die Komplexität der Berechnung eines Nash-Gleichgewichtes für Congestion Spiele untersucht. Bisher war bekannt, dass die Berechnung im Allgemeinen $\mathcal{P}\mathcal{L}\mathcal{S}$ -vollständig ist. Wir haben uns darauf konzentriert die Berechnung eines Nash-Gleichgewichtes mittels einer Minimierung von Rosenthals Potentialfunktion zu untersuchen. Durch eine Reduktion des NOT-ALL-EQUAL-3SAT Problems auf allgemeine Congestion Spiele, konnten wir beweisen, dass die Minimierung von Rosenthals Potentialfunktion stark \mathcal{NP} -schwer ist. Dieses Resultat konnten wir auch für asymmetrische Netzwerk Congestion Spiele zeigen. Hier basierte unser Beweis auf einer Reduktion des ARC-DISJOINT PATHS Problems auf asymmetrische Netzwerk Congestion Spiele. Für symmetrische Netzwerk Congestion Spiele hingegen ist es bekannt, dass ein Nash-Gleichgewicht in polynomieller Zeit berechnet werden kann.

Um die Ergebnisse dieser Arbeit auszubauen, wäre es interessant, weitere Subklassen der Congestion Spiele zu untersuchen. Dabei stellt sich die Frage, ob die Minimierung von Rosenthals Potentialfunktion auch für diese Subklassen \mathcal{NP} -schwer ist. Eine weitere Problemstellung, die in Zukunft genauer untersucht werden kann, ist eine differenziertere Analyse verschiedenartiger Kostenfunktionen der Netzwerk Congestion Spiele. Es gilt herauszufinden, wie schwer es ist, Rosenthals Potentialfunktion für Netzwerk Congestion Spiele für bestimmte Klassen von Kostenfunktionen zu minimieren.

Literaturverzeichnis

- [1] ACKERMANN, HEINER, RÖGLIN, HEIKO und VÖCKING, BERTHOLD: *On the Impact of Combinatorial Structure on Congestion Games*. J. ACM, 55(6):25:1–25:22, Dezember 2008.
- [2] EDMONDS, JACK und KARP, RICHARD M.: *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*. J. ACM, 19(2):248–264, April 1972.
- [3] FABRIKANT, ALEX, PAPADIMITRIOU, CHRISTOS und TALWAR, KUNAL: *The Complexity of Pure Nash Equilibria*. In: *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, Seiten 604–612, New York, NY, USA, 2004. ACM.
- [4] GAREY, MICHAEL R. und JOHNSON, DAVID S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [5] HARKS, TOBIAS, HOEFER, MARTIN, KLIMM, MAX und SKOPALIK, ALEXANDER: *Computing pure Nash and strong equilibria in bottleneck congestion games*. Mathematical Programming, 141(1-2):193–215, 2013.
- [6] JOHNSON, DAVID S., PAPADIMITRIOU, CHRISTOS und YANNAKAKIS, MIHALIS: *How easy is local search?* Journal of computer and system sciences, 37(1):79–100, 1988.
- [7] ORLIN, JAMES B.: *A polynomial time primal network simplex algorithm for minimum cost flows*. Mathematical Programming, 78(2):109–129, 1997.
- [8] RÖGLIN, HEIKO: *Probabilistic Analysis of Algorithms*. Vorlesungsskript, Universität Bonn, Sommersemester 2015. <http://www.i1.informatik.uni-bonn.de/lib/exe/fetch.php?media=lehre:ss15:probanalysis:probanalysis.pdf>.

-
- [9] ROSENTHAL, ROBERT W.: *A class of games possessing pure-strategy Nash equilibria*. International Journal of Game Theory, 2:65–67, 1973.
 - [10] SCHAEFER, THOMAS J.: *The complexity of satisfiability problems*. In: *Proceedings of the tenth annual ACM symposium on Theory of computing*, Seiten 216–226. ACM, 1978.
 - [11] SCHÄFFER, ALEJANDRO A. und YANNAKAKIS, MIHALIS: *Simple Local Search Problems That Are Hard to Solve*. SIAM J. Comput., 20(1):56–87, Februar 1991.
 - [12] WIESE, HARALD: *Entscheidungs- und Spieltheorie*. Springer, 2001.