

# Algorithmische Spieltheorie

Skript<sup>1</sup>

Winter 2019/20

Martin Hofer

<sup>1</sup>Herzlichen Dank an Lisa Wilhelmi für die Aufbereitung und Ausarbeitung des Textes.



# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Strategische Spiele</b>                                       | <b>5</b>  |
| <b>2</b> | <b>Reine Nash-Gleichgewichte</b>                                 | <b>7</b>  |
| 2.1      | Auslastungsspiele (Congestion Games) . . . . .                   | 8         |
| 2.1.1    | Endliche Verbesserungseigenschaft . . . . .                      | 9         |
| 2.1.2    | Konvergenz in Auslastungsspielen . . . . .                       | 11        |
| 2.1.3    | Komplexität . . . . .  | 15        |
| 2.2      | Ordinale Potenziale und Schwache Kreisfreiheit . . . . .         | 18        |
| 2.3      | Zusammenfassung . . . . .  | 25        |
| <b>3</b> | <b>Gemischte Nash-Gleichgewichte</b>                             | <b>27</b> |
| 3.1      | Existenz von gemischten Nash-Gleichgewichten . . . . .           | 29        |
| 3.2      | Komplexität . . . . .  | 32        |
| 3.2.1    | END-OF-LINE . . . . .  | 33        |
| 3.2.2    | Finden von (approximativen) Brouwer-Fixpunkten . . . . .         | 33        |
| 3.3      | Nullsummenspiele . . . . .                                       | 35        |
| 3.3.1    | 2-Spieler Nullsummenspiele . . . . .                             | 36        |
| 3.3.2    | Öffentliche Strategiewahl . . . . .                              | 36        |
| 3.3.3    | Appendix: LP-Dualität . . . . .                                  | 39        |
| 3.4      | Zusammenfassung . . . . .  | 40        |
| <b>4</b> | <b>Korrelierte und grob-korrelierte Gleichgewichte</b>           | <b>41</b> |
| 4.1      | Lernen von grob-korrelierten Gleichgewichten . . . . .           | 43        |
| 4.1.1    | Imitation von Experten . . . . .                                 | 43        |
| 4.1.2    | No-Regret-Algorithmen . . . . .                                  | 44        |
| 4.1.3    | Nullsummenspiele . . . . .                                       | 48        |
| 4.1.4    | Konkave Spiele . . . . .   | 52        |
| 4.2      | Lernen von korrelierten Gleichgewichten . . . . .                | 57        |
| 4.3      | Zusammenfassung . . . . .  | 59        |
| <b>5</b> | <b>Preis der Anarchie und Preis der Stabilität</b>               | <b>61</b> |
| 5.1      | Wardropspiele . . . . .  | 62        |
| 5.1.1    | Existenz und Eindeutigkeit von Wardrop-Gleichgewichten . . . . . | 65        |
| 5.2      | Auslastungsspiele . . . . .                                      | 66        |
| 5.3      | Smoothness . . . . .   | 67        |
| 5.3.1    | Grenzen von Smoothness . . . . .                                 | 69        |
| 5.4      | Preis der Stabilität . . . . .                                   | 70        |
| 5.4.1    | Equal-Sharing-Spiele . . . . .                                   | 70        |
| 5.4.2    | Potenzialspiele . . . . .  | 71        |
| 5.5      | Zusammenfassung . . . . .  | 71        |

|   |            |
|---|------------|
| <b>6 Entwurf anreizkompatibler Mechanismen</b>                      | <b>73</b>  |
| 6.1 Vickrey-Auktion und Vickrey-Clarke-Groves Mechanismen . . . . . | 73         |
| 6.1.1 Rückwärtsauktion . . . . .                                    | 77         |
| 6.2 Direkte Charakterisierung . . . . .                             | 78         |
| 6.3 Single-Parameter Mechanismen . . . . .                          | 79         |
| 6.4 Revelationsprinzip . . . . .                                    | 82         |
| 6.5 Komplexität . . . . .   | 83         |
| 6.5.1 Rucksackauktion . . . . .                                     | 83         |
| 6.6 Ertragsmaximierung . . . . .                                    | 88         |
| 6.7 Zusammenfassung . . . . .                                       | 92         |
| <b>7 Online Auktionen, Sekretäre und Propheten</b>                  | <b>93</b>  |
| 7.1 Sekretäre und Random Order . . . . .                            | 93         |
| 7.2 Prophetische Ungleichung . . . . .                              | 99         |
| 7.3 Zusammenfassung . . . . .                                       | 102        |
| <b>8 Mechanismen als Spiele</b>                                     | <b>105</b> |
| 8.1 Generalized-Second-Price Auktionen . . . . .                    | 105        |
| 8.2 Preis der Anarchie . . . . .                                    | 110        |
| 8.2.1 Zweitpreisauktionen . . . . .                                 | 114        |
| 8.3 Ansteigende Auktionen . . . . .                                 | 115        |
| 8.3.1 Spektrumsauktionen . . . . .                                  | 117        |
| 8.4 Zusammenfassung . . . . .                                       | 118        |
| <b>9 Mechanismen ohne Geld</b>                                      | <b>119</b> |
| 9.1 Wahlen . . . . .  | 119        |
| 9.1.1 Unmöglichkeitsresultate . . . . .                             | 120        |
| 9.2 Strukturierte Präferenzen . . . . .                             | 123        |
| 9.2.1 Präferenzen mit Scheitelpunkt . . . . .                       | 123        |
| 9.2.2 Hausallokation . . . . .                                      | 124        |
| 9.3 Nierenaustausch . . . . .                                       | 127        |
| 9.4 Stabiles Matching . . . . .                                     | 128        |
| 9.5 Zusammenfassung . . . . .                                       | 130        |

# 1 | Strategische Spiele

Spieltheorie befasst sich mit der Modellierung und Analyse von verteilten Systemen mit mehreren Entscheidungsträgern, die unter Wettbewerb agieren. Hierbei werden grundlegende Fragen nach den Konsequenzen, den Anreizen und den sich ergebenden Dynamiken in solchen Umgebungen beantwortet. Die *algorithmische* Spieltheorie legt dabei besonderes Augenmerk auf Aspekte der Komplexität und Approximation.

Das grundlegende Modell in der Spieltheorie ist das strategische Spiel. Hier werden mehrere Spieler betrachtet, die jeweils zwischen verschiedenen Aktionen wählen können. Aus der Kombination der gewählten Aktionen ergibt sich ein Zustand, der für jeden Spieler einen Wert besitzt. Die Spieler haben Präferenzen über diese Zustände.

## Beispiel 1.1. Gefangenendilemma

Ein bekanntes Beispiel ist das Gefangenendilemma. Hier werden zwei Komplizen in unterschiedlichen Räumen wiederholt vernommen, wobei sich beiden die gleichen zwei Optionen darbieten. Entweder die Gefangenen gestehen und verraten den jeweils anderen oder sie schweigen. Wenn beide Komplizen schweigen, schützen sie sich gegenseitig und beide erhalten eine Strafe von 2 (Jahren Haft, Hundert Euro Buße usw.). Einer der beiden kann sich entschließen zu gestehen, sodass er seine eigene Strafe auf 1 verringert und sein Komplize eine Strafe von 5 erhält. Im letzten Szenario gestehen beide Gefangene und erhalten jeweils eine Strafe von 4. Nach jeder Vernehmung wird beiden die aktuelle Strafe mitgeteilt und die Gefangenen können sich neu für eine Option entscheiden.

|   | S | G |
|---|---|---|
| S | 2 | 1 |
| G | 5 | 4 |

Wie sollten sich beide Gefangene entscheiden, um ihre eigene Strafe zu minimieren?

Da die Vernehmungen wiederholt werden, können die Komplizen jeweils auf die Aktion des anderen der letzten Vernehmung reagieren, dabei ist aber keine direkte Absprache möglich. Angenommen beide Gefangenen entscheiden sich zu Beginn zu schweigen, dann erhalten beide die Strafe 2. Wenn einer der beiden nun abweicht und gesteht, kann er seine eigene Strafe von 2 auf 1 verringern. Da der Komplize seine Strafe durch Gestehen ebenfalls minimieren kann, wird dieser sich in dem nächsten Durchgang ebenfalls um entscheiden. Es lohnt sich also in jedem Zustand zu gestehen, obwohl die Gesamtkosten minimiert werden, wenn beide schweigen. ■

**Definition 1.2** (Spiele in Normalform). Ein *strategisches Spiel* in Normalform ist ein Tripel  $(\mathcal{N}, (S_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$ . Dabei ist

- $\mathcal{N}$  die Menge der *Spieler*,  $n = |\mathcal{N}|$ ,
- $S_i$  die Menge der (*reinen*) *Strategien* von Spieler  $i$ ,
- $S = S_1 \times \dots \times S_n$  ist die Menge der Zustände, ein *Zustand*  $s = (s_1, \dots, s_n) \in S$ ,
- $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$  der Zustand  $s$  ohne Strategie  $s_i$ ,
- $c_i : S \rightarrow \mathbb{R}$  die *Kostenfunktion* von Spieler  $i \in \mathcal{N}$ . In Zustand  $s$  hat Spieler  $i$  Kosten  $c_i(s)$ .

Welche Vorhersagen können wir nun über den Ausgang des Spiels treffen? Welche Zustände werden sich ergeben wenn die Spieler ein solches strategisches Spiel spielen? Was sind *stabile* Situationen, von denen kein Spieler mehr abweichen möchte? Im Gefangenendilemma haben wir ein einfaches und überzeugendes Kriterium beobachtet. Dort ist Gestehen eine sog. dominante Strategie, da diese unabhängig von der Strategie des Komplizen immer die geringsten Kosten liefert.

**Definition 1.3** (Dominante Strategie). Eine reine Strategie  $s_i$  ist eine *dominante Strategie* für Spieler  $i \in \mathcal{N}$ , wenn  $c_i(s_i, s_{-i}) \leq c_i(s'_i, s_{-i})$  für jede Strategie  $s'_i \in S_i$  und jedes  $s_{-i}$ .

**Definition 1.4** (Gleichgewichte in dominanten Strategien). Ein Zustand  $s = (s_1, \dots, s_n)$  ist ein *Gleichgewicht in dominanten Strategien*, wenn  $s_i \in S_i$  eine dominante Strategie ist für jeden Spieler  $i \in \mathcal{N}$ .

Es ist leicht einzusehen, dass es Spiele ohne Gleichgewichte in dominanten Strategien gibt. Daher ist dieses Konzept nicht *universell*. Was wäre ein überzeugendes Stabilitätskriterium, wenn kein Gleichgewicht in dominanten Strategien existiert?

Wir betrachten zuerst das folgende dazu orthogonale Konzept.

**Definition 1.5** (Pareto-dominiert). Ein Zustand  $s$  *Pareto-dominiert* einen Zustand  $s'$  (alternativ:  $s$  ist eine *Pareto-Verbesserung* verglichen mit  $s'$ ), wenn  $c_i(s) \leq c_i(s')$  für jeden Spieler  $i \in \mathcal{N}$  und  $c_j(s) < c_j(s')$  für mindestens einen Spieler  $j \in \mathcal{N}$ .

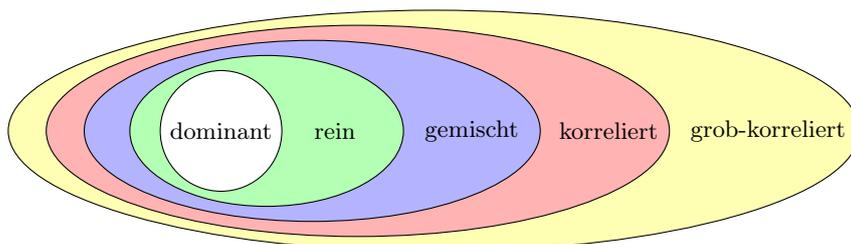
**Definition 1.6** (Pareto-Optimum). Ein Zustand  $s$  heißt *Pareto-Optimum* oder *Pareto-effizient*, wenn es keinen Zustand gibt, der  $s$  Pareto-dominiert.

In einem Pareto-Optimum kann kein Spieler seine Kosten strikt verringern, ohne dass sich die Kosten eines anderen Spielers strikt erhöhen. Im Gefangenendilemma gibt es die drei Pareto-Optima (S,S), (S,G) und (G,S).

Das Pareto-Optimum ist universell, d.h. jedes strategische Spiel hat ein Pareto-Optimum! Der Zustand mit der geringsten Kostensumme ist zwangsweise ein Pareto-Optimum. Wenn alle Spieler abweichen, erhöht sich die Summe und daher muss sich mindestens ein Spieler verschlechtern. Das Pareto-Optimum ist also in diesem Sinne kein Gleichgewichtskonzept für eigensinnige Spieler, da es verbessernde Abweichungen eines Spielers erlaubt (es verschlechtert sich dabei aber mindestens ein anderer Spieler).

Da viele Spiele kein Gleichgewicht in dominanten Strategien besitzen, ist dieses Konzept im Allgemeinen zu speziell. Auch die Pareto-Optimalität ist oft, wie im Gefangenendilemma, für eine Analyse wenig aussagekräftig und beschreibt keine stabilen Zustände. Im Gegensatz dazu suchen wir nach allgemeineren Formen von Gleichgewichten, die möglichst in jedem Spiel enthalten und effizient berechenbar sind. Ist ein Zustand ein Gleichgewicht, möchte auch im Nachhinein kein Spieler von seiner eigenen Strategie abweichen.

In den folgenden Kapiteln werden wir vier weitere Gleichgewichtstypen in strategischen Spielen kennenlernen: reine, gemischte, korrelierte und grob-korrelierte Gleichgewichte. Diese bilden eine Hierarchie, d.h. jedes reine Gleichgewicht ist auch ein gemischtes, jedes gemischte Gleichgewicht auch ein korreliertes, etc. Daneben werden wir uns mit Fragen der Berechenbarkeit und Komplexität befassen, sowie soziale Kosten von Gleichgewichten betrachten und mit Hilfe von Approximationsfaktoren und dem Preis der Anarchie beschränken.



## 2 | Reine Nash-Gleichgewichte

In den meisten strategischen Spielen lässt sich die Entwicklung nicht durch globale Präferenzen der Spieler bestimmen. Dafür wird in diesem Kapitel ein Gleichgewicht vorgestellt, das eine stabile Kollektion von lokalen Präferenzen darstellt. Hierbei wählt jeder Spieler eine Strategie, sodass er durch alleiniges Abweichen keine Verbesserung der eigenen Kosten erzielen kann. Durch diese Eigenschaft ist ein Zustand im Gleichgewicht stabil – für jeden Spieler gilt, dass wenn kein anderer Spieler abweicht, er selbst auch nicht von seiner Strategie abweichen möchte.

### Beispiel 2.1. Kampf der Geschlechter

Ein Pärchen plant das kommende Wochenende, wobei die zwei möglichen Aktivitäten „Zeil“ und „Eintracht“ zur Wahl stehen. Die beiden präferieren je eine der Möglichkeiten, bewerten eine gemeinsame Aktivität aber deutlich höher als alleine ihrer Präferenz nachzugehen.

|           | Zeil   | Eintracht |
|-----------|--------|-----------|
| Zeil      | 1<br>2 | 6<br>6    |
| Eintracht | 5<br>5 | 2<br>1    |

Wie wird das Pärchen das Wochenende verbringen? In diesem Beispiel gibt es keine dominante Strategie, die die Spieler immer verfolgen möchten, stattdessen wählen hier beide in Abhängigkeit des anderen. Dabei werden sich beide Partner für die gleiche Aktivität entscheiden und möchten beide nicht mehr von dieser unilateral abweichen. ■

**Definition 2.2** (Beste Antwort). Eine Strategie  $s_i$  heißt eine *beste Antwort* auf eine Kollektion von Strategien  $s_{-i}$ , wenn  $c_i(s_i, s_{-i}) \leq c_i(s'_i, s_{-i})$  für alle  $s'_i \in S_i$ .

Beachte:  $s_i$  dominante Strategie  $\Leftrightarrow s_i$  beste Antwort auf alle  $s_{-i}$ .

**Definition 2.3** (Reines Nash-Gleichgewicht). Ein Zustand  $s = (s_1, \dots, s_n)$  heißt *Nash-Gleichgewicht in reinen Strategien (oder reines Nash-Gleichgewicht)*, wenn  $s_i$  eine beste Antwort auf die anderen Strategien  $s_{-i}$  ist, für jeden Spieler  $1 \leq i \leq n$ .

Ein Nash-Gleichgewicht ist eine Kollektion von lokalen Präferenzen im Spiel. Kein Spieler hat einen Anreiz seine Strategie zu ändern, d.h. ein reines Nash-Gleichgewicht ist stabil gegen unilaterale Abweichungen. Im gegebenen Beispiel bilden die Zustände (Zeil, Zeil) und (Eintracht, Eintracht) je ein reines Nash-Gleichgewicht.

Ist das reine Nash-Gleichgewicht ein universelles Lösungskonzept, das in jedem Spiel existiert? Nein, es gibt durchaus Spiele ohne reine Nash-Gleichgewichte (siehe Beispiel 3.1)! Daher betrachten wir im Rest des Kapitels prominente Klassen von Spielen, in denen reine Nash-Gleichgewichte existieren. In diesen Spielen ist das reine Gleichgewicht ein attraktives Konzept, um die Zustände erfassen, die sich aus dem Wettbewerb der Spieler ergeben.

## 2.1 Auslastungsspiele (Congestion Games)

Diese Klasse von Spielen modelliert Szenarien wie die Verkehrsnutzung, Netzwerkauslastung oder Nutzung von mehreren Prozessoren. Gegeben ist eine Menge an Ressourcen, die häufig von vielen Spielern verwendet wird, dabei verändert sich die Verzögerung einer Ressource bei steigender Nutzung.

**Definition 2.4** (Auslastungsspiel). Ein *Auslastungsspiel* ist ein Tupel  $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$  mit

- $\mathcal{N} = \{1, \dots, n\}$ , Menge der Spieler,
- $\mathcal{R} = \{1, \dots, m\}$ , Menge der Ressourcen,
- $\Sigma_i \subseteq 2^{\mathcal{R}}$ , Strategieraum von Spieler  $i$ ,
- $d_r : \{1, \dots, n\} \rightarrow \mathbb{Z}$ , Latenz- oder Verzögerungsfunktion von Ressource  $r$ .

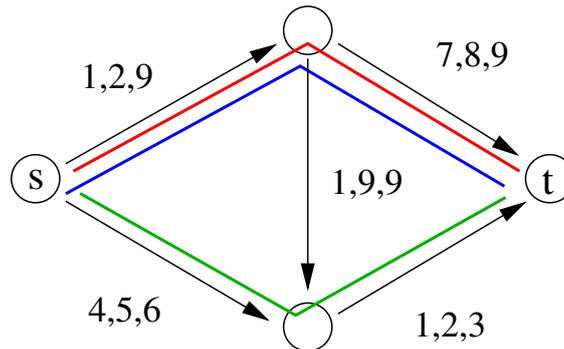
Für jeden Zustand  $S = (S_1, \dots, S_n) \in \Sigma_1 \times \dots \times \Sigma_n$  sei

- $n_r =$  Anzahl der Spieler mit  $r \in S_i$ ,
- $d_r(n_r) =$  Latenz oder Verzögerung der Ressource  $r$ ,
- $\delta_i(S) = \sum_{r \in S_i} d_r(n_r) =$  Latenz oder Verzögerung von Spieler  $i$ .

Die *Kosten* von Spieler  $i$  in Zustand  $S$  sind  $c_i(S) = \delta_i(S)$ . Die Spieler wollen ihre Kosten verringern.

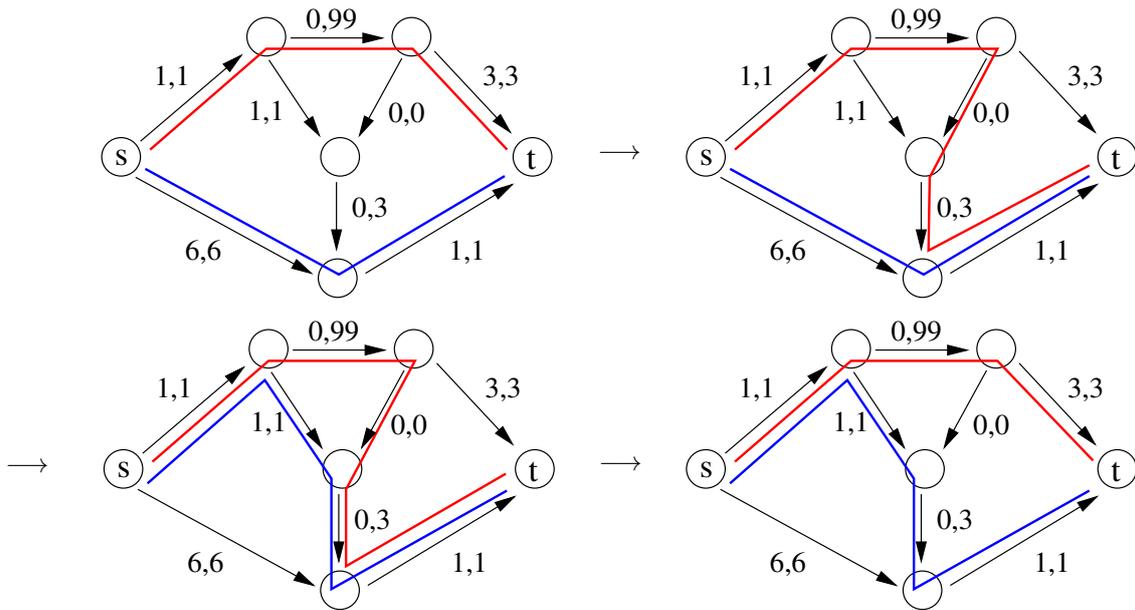
**Beispiel 2.5.** *Netzwerkauslastungsspiele*

Sei  $G = (V, E)$  ein gerichteter Graph mit den Ressourcen  $E$ , wobei jede Kante  $e \in E$  eine Latenzfunktion  $d_e : \{1, \dots, n\} \rightarrow \mathbb{Z}$  besitzt. Spieler  $i$  möchte einen Weg wählen mit kleinster Latenz zwischen seinem Startknoten  $s_i$  und dem Endknoten  $t_i$ .



In diesem Beispiel ist  $\mathcal{N} = \{1, 2, 3\}$ ,  $\mathcal{R} = E$ ,  $\Sigma_i =$  Menge der s-t-Wege. Dieses Spiel ist *symmetrisch*: Alle Spieler haben die gleiche Strategiemenge, und wenn wir in einem Zustand die Strategien der Spieler permutieren, dann ergeben sich die Kosten in ebenso permutierter Weise.

Das folgende Beispiel verdeutlicht, wie ein reines Nash-Gleichgewicht zustande kommen kann. Dazu werden ein gerichteter Graph und zwei Spieler betrachtet, die abwechselnd den s-t-Pfad mit der geringsten Latenz wählen.



Da sich kein Spieler durch alleiniges Abweichen verbessern kann, bildet der letzte Zustand ein reines Nash-Gleichgewicht. ■

Im gegebenen Beispiel kann ein reines Nash-Gleichgewicht durch eine Folge von Verbesserungsschritten erreicht werden. Existiert in jedem Auslastungsspiel ein reines Nash-Gleichgewicht und kann dieses ggf. immer durch Verbesserungsfolgen erreicht werden? Wie viele Schritte benötigt solch eine Folge?

### 2.1.1 Endliche Verbesserungseigenschaft

**Korollar 2.6.** *Jedes Auslastungsspiel hat ein reines Nash-Gleichgewicht.*

Das Korollar folgt direkt aus dem folgenden Satz:

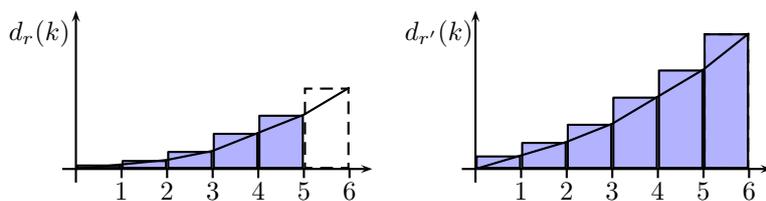
**Satz 2.7** (Rosenthal 1973). *In einem Auslastungsspiel ist jede Folge von Verbesserungsschritten endlich.*

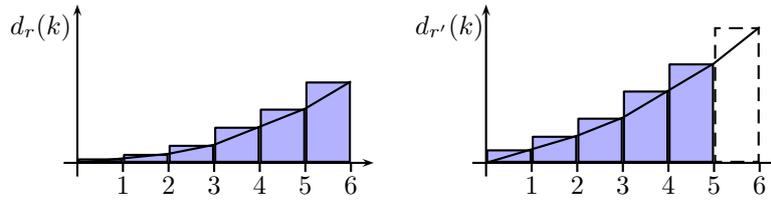
*Beweis (Satz 2.7).* Rosenthals Analyse basiert auf einer Potenzialfunktion. Für jeden Zustand  $S$  sei

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

Diese Funktion heißt *Rosenthals Potenzialfunktion*.

**Lemma 2.8.** *Sei  $S$  ein beliebiger Zustand. Wenn wir von  $S$  zu Zustand  $S'$  wechseln mit einem Verbesserungsschritt von Spieler  $i$ , in dem seine Kosten um  $\Delta > 0$  verringert werden, dann gilt  $\Phi(S') = \Phi(S) - \Delta$ .*





Im Bild ist der Wert der Potenzialfunktion die blaue Fläche. Wenn ein Spieler von  $r'$  zu  $r$  wechselt, verändern sich seine Kosten genau wie der Wert der Potenzialfunktion.

*Beweis.* Das Potenzial  $\Phi(S)$  kann berechnet werden, indem Spieler nacheinander in beliebiger Ordnung in das Spiel eingefügt und die Latenzen zum Zeitpunkt des Einfügens summiert werden. O.B.d.A. sei Spieler  $i$  der letzte Spieler, den wir einfügen, wenn wir  $\Phi(S)$  berechnen. Für diesen Spieler addieren wir seine tatsächlichen Kosten in Zustand  $S$  zu  $\Phi(S)$ . Wenn wir uns von  $S$  zu  $S'$  bewegen, dann sinken die Kosten von  $i$  um  $\Delta$ . Folglich sinkt das Potenzial  $\Phi$  auch genau um  $\Delta$ .  $\square$

Das Lemma zeigt, dass  $\Phi$  ein *exaktes Potenzial* ist, d.h. wenn ein einzelner Spieler seine Latenz um einen Wert  $\Delta > 0$  verringert, dann sinkt  $\Phi$  um genau den gleichen Wert  $\Delta$ . Daneben stellen wir fest:

- i) Die Latenzwerte sind ganze Zahlen. Bei jeder Verbesserung gilt  $\Delta \geq 1$ .
- ii) Für jeden Zustand  $S$  gilt  $\Phi(S) \leq \sum_{r \in \mathcal{R}} \sum_{i=1}^n |d_r(i)|$ .
- iii) Für jeden Zustand  $S$  gilt  $\Phi(S) \geq -\sum_{r \in \mathcal{R}} \sum_{i=1}^n |d_r(i)|$ .

Daraus folgt, dass die Anzahl Verbesserungsschritte höchstens  $2 \cdot \sum_{r \in \mathcal{R}} \sum_{i=1}^n |d_r(i)|$  beträgt und endlich ist.  $\square$  (Satz)

### Exakte Potenzialspele

**Definition 2.9** (Exaktes Potenzialspiel). Ein strategisches Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$  besitzt eine *exakte Potenzialfunktion*  $\Phi: \Sigma \rightarrow \mathbb{R}$ , wenn für jedes  $i \in \mathcal{N}$ , für jedes  $S_{-i} \in \Sigma_{-i}$ , und für jede  $S_i, S'_i \in \Sigma_i$  gilt:

$$c_i(S_i, S_{-i}) - c_i(S'_i, S_{-i}) = \Phi(S_i, S_{-i}) - \Phi(S'_i, S_{-i}) .$$

Ein solches Spiel nennen wir (*exaktes*) *Potenzialspiel*.

**Beobachtung:** Sei  $\Gamma$  ein exaktes Potenzialspiel. Dann hat  $\Gamma$  die *endliche Verbesserungseigenschaft* (Engl: *finite improvement property*), und daher existiert ein reines Nash-Gleichgewicht.

Rosenthals Potenzialfunktion ist eine exakte Potenzialfunktion. Daher gilt:

**Korollar 2.10.** *Jedes Auslastungsspiel ist ein exaktes Potenzialspiel.*

In gewisser Weise gilt auch die andere Richtung:

**Satz 2.11** (Monderer, Shapley, 1996). *Jedes exakte Potenzialspiel ist "isomorph" zu einem Auslastungsspiel.*

Für jedes exakte Potenzialspiel gibt es ein anderes Spiel mit den gleichen Spielern, Strategien und Kosten. Das andere Spiel basiert auf geschickt zusammengestellten Ressourcen und Latenzen, so dass die Strategien als Teilmengen von Ressourcen interpretiert werden können und die Kosten sich als Summe über deren Latenzen ergeben. In diesem Sinne gibt es für jedes exakte Potenzialspiel eine Representation als "isomorphes" Auslastungsspiel.

### 2.1.2 Konvergenz in Auslastungsspielen

Durch den Satz von Rosenthal folgt die Existenz von reinen Nash-Gleichgewichten in Auslastungsspielen. Ebenfalls wird deutlich, dass die Spieler dieses Gleichgewicht durch eine endliche Verbesserungsfolge erreichen können. Wie viele Verbesserungsschritte werden benötigt, um ein reines Nash-Gleichgewicht zu erreichen?

Der **Übergangsgraph** eines Auslastungsspiels  $\Gamma$  enthält einen Knoten für jeden Zustand  $S$  und eine gerichtete Kante  $(S, S')$  wenn  $S'$  von  $S$  mit einem Verbesserungsschritt eines einzelnen Spielers erreicht werden kann. Der **Beste-Antwort-Übergangsgraph** enthält nur Kanten für Verbesserungsschritte, die besten Antworten entsprechen.

Eine Folge von Verbesserungsschritten (bzw. besten Antworten) entspricht einem Weg im Übergangsgraphen (bzw. Beste-Antwort-Übergangsgraphen). Die Senken dieses Graphen sind die Nash-Gleichgewichte von  $\Gamma$ . Die Anzahl der Knoten (Zustände) kann bis zu  $2^{mn}$  betragen. Daher kann es Wege mit **exponentieller Länge** geben.

#### Singleton-Auslastungsspiele

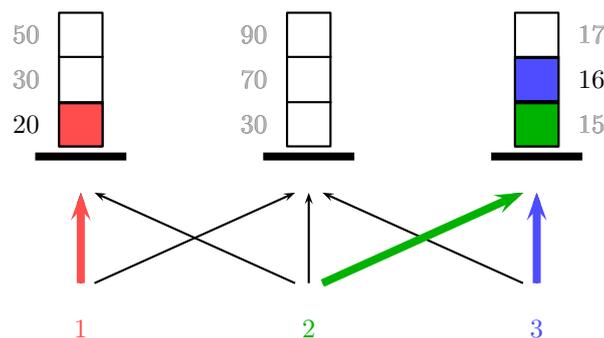
**Definition 2.12** (Singleton-Auslastungsspiel). Ein Auslastungsspiel heißt *singleton*, wenn für jeden Spieler  $i \in \mathcal{N}$  und jede Strategie  $R \in \Sigma_i$  gilt  $|R| = 1$ .

In Worten: Jeder Spieler möchte nur eine Ressource aus einer für ihn zulässigen Teilmenge von Ressourcen auswählen.

Obwohl dies eine relativ starke Einschränkung des Strategieraumes ist, gibt es immer noch bis zu  $m^n$  unterschiedliche Zustände in einem Spiel dieser Art.

#### Beispiel 2.13. Serverfarm

Betrachte eine "Server-Farm" mit drei Servern  $a, b, c$  (Ressourcen) und drei Spielern 1, 2, 3. Jeder der Spieler möchte einen Task auf genau einem Server berechnen lassen und wählt diesen strategisch nach der resultierenden Bearbeitungszeit. Je nachdem wie sehr ein Server ausgelastet ist (von wie vielen Spielern er gewählt wurde), steigt seine Bearbeitungszeit für die Tasks. Den Spielern steht jeweils eine Teilmenge der Server zur Auswahl, diese wird durch die Pfeile gekennzeichnet.



Dieses Beispiel stellt ein reines Nash-Gleichgewicht dar, dabei hat Spieler 1 die Kosten 20, Spieler 2 und 3 je die Kosten 16. Durch unilaterales Abweichen kann keiner seine Kosten verringern. ■

**Satz 2.14.** In Singleton-Auslastungsspielen hat jede Verbesserungsfolge eine Länge von  $O(n^2m)$ .

*Beweis.* Die Idee ist wie folgt: Wir ersetzen Latenzwerte durch beschränkte, positive ganze Zahlen ohne die Präferenzen der Spieler zu verändern. Damit ergibt sich eine obere Schranke auf den

Maximalwert des Potenzials. Durch ganzzahlige Latenzen sinkt das Potenzial im Verbesserungsschritt um mind. 1. Daher ist die Länge jeder Verbesserungsfolge beschränkt durch maximalen Potenzialwert.

Genauer gesagt, sortiere die Menge der Latenzwerte  $\{d_r(k) \mid r \in \mathcal{R}, 1 \leq k \leq n\}$  in nicht-fallender Reihenfolge. Wir definieren alternative, neue Latenzfunktionen:

$$\bar{d}_r(k) := \text{Position von } d_r(k) \text{ in Sortierung.}$$

Die sortierte Menge von Latenzen aus dem obigen Beispiel ist

$$15, 16, 17, 20, 30, 50, 70, 90.$$

Daher ergeben sich die alten und neuen Latenzfunktionen als

$$\begin{aligned} d_a(1, 2, 3) &= (20, 30, 50) & \bar{d}_a(1, 2, 3) &= (4, 5, 6) \\ d_b(1, 2, 3) &= (30, 70, 90) & \bar{d}_b(1, 2, 3) &= (5, 7, 8) \\ d_c(1, 2, 3) &= (15, 16, 17) & \bar{d}_c(1, 2, 3) &= (1, 2, 3) \end{aligned}$$

Die neue Latenz von Spieler  $i$  auf Ressource  $r$  in Zustand  $S$  ist  $\bar{d}_i(S) = \bar{d}_r(n_r(S))$ .

**Beobachtung:**

Seien  $S$  und  $S'$  zwei Zustände, so dass  $(S, S')$  einen Verbesserungsschritt darstellt für einen Spieler  $i$  mit originalen Latenzen. Dann ist  $(S, S')$  auch ein Verbesserungsschritt für  $i$  mit neuen Latenzen.

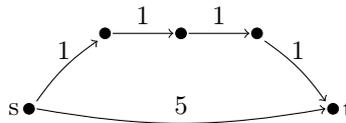
Rosenthals Potenzialfunktion mit neuen Latenzen ist beschränkt durch:

$$\bar{\Phi}(S) = \sum_{r \in \mathcal{R}} \sum_{k=1}^{n_r(S)} \bar{d}_r(k) \leq \sum_{r \in \mathcal{R}} \sum_{k=1}^{n_r(S)} n m \leq n^2 m .$$

Es gilt  $\bar{\Phi} \geq 1$ . Daneben fällt  $\bar{\Phi}$  um mind. 1 in jedem Schritt. Daher beträgt die Länge jeder Verbesserungsfolge höchstens  $n^2 m$ . □

Warum lässt sich der Beweis nicht auf allgemeine Auslastungsspiele übertragen?

Das gegebene Netzwerk besitzt zwei verschiedene s-t-Pfade, wobei der obere Pfad weniger Kosten besitzt. Nach einer Rangersetzung würde der untere Pfad geringere Kosten verursachen.



Für Singleton-Spiele konnten wir zeigen, dass **jede Verbesserungsfolge eine Länge polynomiell in  $n$  und  $m$  hat**. Diese Schranke gilt für jede beliebige Verbesserungsfolge, bei denen der wechselnde Spieler mit seiner neuen Strategie seine Kosten strikt verringert (nicht nur beste Antworten).

Dieses Resultat lässt sich noch verallgemeinern auf sog. *Matroidspiele*. Hier besteht die Strategiemenge jedes Spielers aus den Basen eines Matroids über die Ressourcen. In solchen Spielen kann man für **Folgen von besten Antworten** zeigen, dass sie eine **Länge polynomiell in  $n$  und  $m$  haben**.

Im Allgemeinen gibt es aber für jedes  $n \in \mathbb{N}$  mindestens ein Spiel mit

- $O(n)$  Spielern und  $O(n)$  Ressourcen,
- nicht-negativen und monotonen Latenzfunktionen, und

- einem ausgewiesenen Startzustand  $S$ ,

so dass **jede Verbesserungsfolge** von  $S$  zu einem reinen Nash-Gleichgewicht eine **Länge exponentiell in  $n$**  hat.

### Approximative Gleichgewichte

Wie verhält sich die Konvergenzzeit, wenn die Spieler auch mit einer Strategie zufrieden sind, die *fast* so gut ist wie die beste Antwort?

Wir nutzen  $0 \leq \varepsilon < 1$  für eine ‘‘Fehlertoleranz’’ der Spieler.

**Definition 2.15** ( $\varepsilon$ -Verbesserung). Eine beste Antwort  $s'_i$  heißt eine  $\varepsilon$ -Verbesserung für Spieler  $i$  im Zustand  $s$ , wenn  $c_i(s) - c(s'_i, s_{-i}) > \varepsilon \cdot c_i(s)$ . Das Paar  $(s, (s'_i, s_{-i}))$  heißt  $\varepsilon$ -Verbesserungsschritt.

**Definition 2.16** (Approximatives Nash-Gleichgewicht). Ein Zustand  $s$  heißt (*reines*)  $\varepsilon$ -approximatives Nash-Gleichgewicht, wenn kein Spieler  $i \in \mathcal{N}$  eine  $\varepsilon$ -Verbesserung hat.

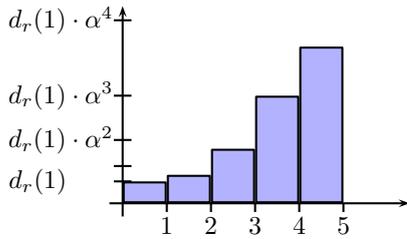
Eine  $\varepsilon$ -Verbesserung für Spieler  $i$  im Zustand  $s$  verringert die Kosten um mehr als einen  $\varepsilon$ -Bruchteil seiner momentanen Kosten. Im  $\varepsilon$ -approximativen Gleichgewicht  $s$  kann kein Spieler  $i$  seine Kosten  $c_i(s)$  um mehr als  $\varepsilon \cdot c_i(s)$  reduzieren, wenn er alleine zu einer anderen Strategie abweicht.

Wir betrachten Folgen von  $\varepsilon$ -Verbesserungsschritten in einer großen Klasse von Spielen mit einer natürlichen Bedingung auf die Kostenfunktionen.

**Definition 2.17** (Symmetrisches Auslastungsspiel). Ein Auslastungsspiel heißt *symmetrisch*, wenn  $\Sigma_i = \Sigma_j$  für alle Spieler  $i, j \in \mathcal{N}$ .

**Definition 2.18** ( $\alpha$ -sprungbeschränkte Latenzfunktion). Sei  $\alpha \geq 1$  und  $d_r$  eine positive, monoton steigende Latenzfunktion.  $d_r$  ist  $\alpha$ -sprungbeschränkt, wenn für alle  $k \geq 2$  gilt

$$d_r(k) \leq \alpha \cdot d_r(k - 1) .$$



Durch Hinzufügen eines weiteren Spielers auf Resource  $r$  erhöhen sich die Kosten aller Spieler auf  $r$  nur um den Faktor  $\alpha$ . Damit gilt insbesondere  $d_r(k) \leq d_r(1) \cdot \alpha^{k-1}$ . Wir bezeichnen mit  $\Phi^* = \min_{S \in \Sigma} \Phi(S)$  den kleinsten Wert von Rosenthals Potenzialfunktion für einen Zustand im Spiel. Für positive Latenzfunktionen in natürlichen Zahlen gilt offensichtlich  $\Phi^* \geq n$ .

**Satz 2.19** (Chien, Sinclair, 2011). Sei  $\Gamma$  ein symmetrisches Auslastungsspiel mit  $\alpha$ -sprungbeschränkten Latenzfunktionen in  $\mathbb{N}$ . Für jeden Zustand  $S$  gibt es eine Folge von  $\varepsilon$ -Verbesserungsschritten mit der Länge

$$O\left(\frac{n\alpha}{\varepsilon} \cdot \log\left(\frac{\Phi(S)}{\Phi^*}\right)\right) = O\left(\frac{n\alpha}{\varepsilon} \cdot \log\left(m \cdot \max_{r \in \mathcal{R}} d_r(n)\right)\right) ,$$

die in einem  $\varepsilon$ -approximativen Nash-Gleichgewicht endet.

*Beweis.* Wir konstruieren die Folge mit einer *relativen  $\varepsilon$ -Nash-Dynamik*:

Sei  $I(S)$  die Menge der Spieler mit  $\varepsilon$ -Verbesserungen in  $S$ , und sei  $S'_i$  eine  $\varepsilon$ -Verbesserung für  $i \in I(S)$ . Der abweichende Spieler  $i^* \in I(S)$  wird gewählt als ein Spieler mit maximaler relativer Kostenverbesserung:

$$i^* \in \arg \max_{i \in I(S)} \frac{c_i(S) - c_i(S'_i, S_{-i})}{c_i(S)} .$$

Betrachte nun einen beliebigen anderen Spieler  $j \neq i^*$  im Zustand  $S$ . Da das Spiel symmetrisch ist, hätte auch  $j$  zu  $S'_{i^*}$  abweichen können. Wenn  $j$  anstelle von  $i^*$  nach  $S'_{i^*}$  abweicht, und weil die Latenzfunktionen  $\alpha$ -sprungbeschränkt sind, wären seine Kosten höchstens:

$$c_j(S'_{i^*}, S_{-j}) \leq \alpha \cdot c_{i^*}(S'_{i^*}, S_{-i^*}) .$$

Die relative  $\varepsilon$ -Dynamik hat aber Spieler  $i^*$  gewählt, daher:

$$\frac{c_{i^*}(S) - c_{i^*}^*(S'_{i^*}, S_{-i^*})}{c_{i^*}(S)} \geq \frac{c_j(S) - c_j(S'_{i^*}, S_{-j})}{c_j(S)} \geq \frac{c_j(S) - \alpha c_{i^*}(S'_{i^*}, S_{-i^*})}{c_j(S)} .$$

Eine Umformung des ersten und dritten Bruchs ergibt:  $c_{i^*}(S) \geq c_j(S)/\alpha$ .

Der *wechselnde Spieler*  $i^*$  in  $S$  ist also – bis auf einen Faktor  $\alpha$  – auch der Spieler mit *maximalen Kosten*:  $c_{i^*}(S) \geq \max_j c_j(S)/\alpha$ .

Damit wissen wir:

$$\begin{aligned} \Phi(S) - \Phi(S'_{i^*}, S_{-i^*}) &\geq \varepsilon \cdot c_{i^*}(S) \geq \frac{\varepsilon}{\alpha} \cdot \max_j c_j(S) \\ &\geq \frac{\varepsilon}{\alpha} \cdot \frac{1}{n} \cdot \sum_j c_j(S) \geq \frac{\varepsilon}{n\alpha} \cdot \Phi(S) \end{aligned}$$

Das Potenzial verringert sich also in jedem Verbesserungsschritt der relativen  $\varepsilon$ -Dynamik um einen Faktor von  $\varepsilon/(n\alpha)$ . Seien  $S^0, S^1, S^2, \dots$  die resultierende Folge von Zuständen (mit  $S = S^0$ ), dann gilt

$$\Phi(S^k) \leq \Phi(S^0) \cdot \left(1 - \frac{\varepsilon}{n\alpha}\right)^k .$$

Wir drücken nun  $k$  als Vielfaches von  $n\alpha/\varepsilon$  aus und beobachten

$$\Phi(S) \cdot \left(1 - \frac{\varepsilon}{n\alpha}\right)^k = \Phi(S) \cdot \left(1 - \frac{\varepsilon}{n\alpha}\right)^{\ell \cdot n\alpha/\varepsilon} \leq \Phi(S) \cdot \frac{1}{e^\ell} ,$$

da  $(1-x)^{1/x} \leq (e^{-x})^{1/x} = 1/e$  für alle  $0 < x \leq 1$ . Wenn  $\ell$  so groß wird, dass

$$\Phi(S)/e^\ell \leq \Phi^*$$

gilt, dann kann es keine Verbesserung mehr geben. Daraus folgt, dass bei

$$\ell \geq \ln \frac{\Phi(S)}{\Phi^*}$$

keine Verbesserung mehr stattfinden kann. Daher gilt, dass die Anzahl  $k$  der Schritte der relativen  $\varepsilon$ -Dynamik von Zustand  $S$  bis zum  $\varepsilon$ -approximativen Nash-Gleichgewicht höchstens

$$k = \frac{n\alpha}{\varepsilon} \cdot \ell \leq \frac{n\alpha}{\varepsilon} \cdot \ln \frac{\Phi(S)}{\Phi^*}$$

betragen kann. □

Dieses Resultat gilt nicht mehr in (1) *asymmetrischen* Spielen mit  $\alpha$ -sprungbeschränkten Funktionen oder (2) *symmetrischen* Spielen, in denen die Latenzfunktionen *nicht*  $\alpha$ -sprungbeschränkt sind.

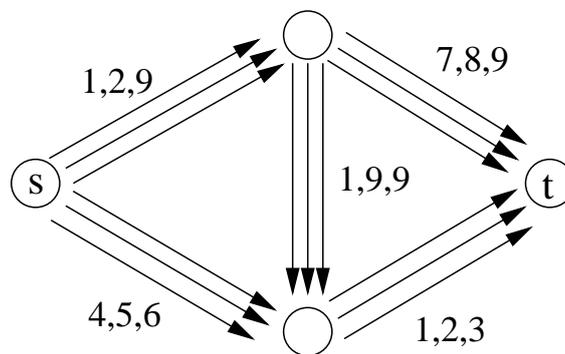
### 2.1.3 Komplexität

Wir untersuchen die Komplexität des Auffindens von reinen Nash-Gleichgewichten in Klassen von Auslastungsspielen. Dabei beschränken wir uns auf Spiele mit nicht-fallenden Latenzfunktionen. Betrachtet man ausschließlich symmetrische Netzwerkspiele, gibt es sogar hier Spiele mit Startzuständen, so dass jede Verbesserungsfolge zu einem reinen Nash-Gleichgewicht exponentielle Länge hat. Daher sind Verbesserungsfolgen keine **effizienten** (d.h. Polynomzeit) Algorithmen zur Berechnung von reinen Nash-Gleichgewichten in diesen Spielen.

Es gibt allerdings andere Algorithmen, die ein Nash-Gleichgewicht in polynomieller Zeit finden...

#### Effizienter Algorithmus durch Reduktion auf Netzwerkfluß

Im gegebenen Graphen wird jede Kante ersetzt durch  $n$  parallele Kanten mit (jeweils) Kapazität 1. Dabei produziert die  $i$ -te Kopie von Kante  $e$  die Kosten  $d_e(i)$ ,  $1 \leq i \leq n$ .



Wir berechnen einen Min-Cost-Flow, d.h. einen Netzwerkfluß mit Wert  $n$  von  $s$  nach  $t$ , der die Kosten der benutzten Kantenkopien minimiert. Die optimale Lösung minimiert Rosenthals Potenzialfunktion und liefert damit ein reines Nash-Gleichgewicht.

#### Komplexitätsklasse PLS

Rosenthals Potenzialfunktion erlaubt eine Interpretation von Auslastungsspielen als lokales Suchproblem: Nash-Gleichgewichte sind lokale Optima bzgl. der Potenzialfunktion.

Wie schwierig ist die Berechnung von lokalen Optima?

Dazu betrachten wir die Komplexitätsklasse Polynomial Local Search (PLS).

**Definition 2.20** (PLS (Polynomial Local Search)). PLS enthält Suchprobleme mit einer Zielfunktion und einer gegebenen Nachbarschaftsbeziehung  $\Gamma$ . Es muss einen Polynomialzeit-Algorithmus geben, der für jede Lösung  $s$ ,

- eine Lösung in  $\Gamma(s)$  mit besserem Zielfunktionswert berechnet, oder
- wenn  $s$  ein lokales Optimum ist, dies korrekt ausgibt.

Beispiele für Probleme in PLS:

- FLIP (Evaluation von Schaltkreisen mit Flip-Nachbarschaft)
- TSP with 2-Opt-Nachbarschaft
- Pos-NAE- $k$ -SAT mit Flip-Nachbarschaft
- Max-Cut mit Flip-Nachbarschaft
- Auslastungsspiele bzgl. Verbesserungsschritte

**Definition 2.21** (Positive Not-All-Equal  $k$ -Sat (Pos-NAE- $k$ -SAT)). Als Eingabe dient eine Formel aus  $n$  binären Variablen  $x_1, \dots, x_n$ . Beschrieben wird die Problemstellung durch  $m$  Klauseln  $c_1, \dots, c_m$  mit jeweils  $k$  positiven Literalen, und  $m$  Gewichten  $w_1, \dots, w_m$ .

Eine Klausel  $c_i$  gilt als *erfüllt* durch eine Zuweisung  $A \in \{0, 1\}^n$ , wenn ihre Literale nicht alle den gleichen Wert haben. Der *Wert* einer Zuweisung ist die Summe der Gewichte der erfüllten Klauseln. Des Weiteren sind die Zuweisungen  $A$  und  $A'$  *benachbart*, wenn sie sich in genau einer Position unterscheiden.

Die Aufgabe besteht darin, ein *lokales Optimum*, d.h. eine Zuweisung ohne benachbarte Zuweisung mit höherem Wert, zu finden.

**Beispiel 2.22.** *Pos-NAE-3-SAT*

$$c_1 = (x_1, x_2, x_3); c_2 = (x_1, x_2, x_4); c_3 = (x_1, x_2, x_5); c_4 = (x_3, x_4, x_5)$$

$$w_1 = 100; w_2 = 110; w_3 = 120; w_4 = 100$$

Ein Beispiel für ein **lokales Optimum** ist die Zuweisung

$$x_1 = 0; x_2 = 0; x_3 = 1; x_4 = 1; x_5 = 1$$

mit Wert 330. Jede der fünf benachbarten Zuweisungen hat einen Wert von höchstens 330. ■

**Definition 2.23** (Max-Cut). Gegeben sei ein ungerichteter Graph  $G = (V, E)$  mit Kantengewichten  $w : E \rightarrow \mathbb{N}$ . Ein *Schnitt* zerteilt  $V$  in zwei Mengen *Links* und *Rechts*. Weiterhin sind zwei Schnitte *benachbart*, wenn der eine Schnitt aus dem anderen Schnitt durch Verschieben genau eines Knotens von *Links* nach *Rechts* oder umgekehrt konstruierbar ist. Der *Wert* eines Schnittes ist das Gesamtgewicht aller Kanten mit einem Endpunkt in *Links* und einem Endpunkt in *Rechts*. Gesucht ist ein *lokales Optimum*, d.h. einen Schnitt ohne benachbarten Schnitt mit höherem Wert.

**Definition 2.24** (PLS-Reduktion). Gegeben zwei PLS-Probleme  $\Pi_1$  und  $\Pi_2$ , finde eine Abbildung der Instanzen von  $\Pi_1$  auf die Instanzen von  $\Pi_2$ , so dass

- die Abbildung in polynomieller Zeit berechnet werden kann,
- die lokalen Optima von  $\Pi_1$  auf die lokalen Optima von  $\Pi_2$  abgebildet werden und
- aus jedem lokalen Optimum von  $\Pi_2$  kann man ein lokales Optimum von  $\Pi_1$  in polynomieller Zeit errechnen.

**Satz 2.25.** *Pos-NAE-3-SAT  $\leq_{PLS}$  Pos-NAE-2-SAT.*

*Beweis.* Für jede 3-Klausel  $(x_1, x_2, x_3)$  mit Gewicht  $w$  erstelle die drei 2-Klauseln  $(x_1, x_2), (x_1, x_3), (x_2, x_3)$ , jede mit Wert  $w/2$ .

Der Wert einer Zuweisung in der 2-SAT Instanz ist identisch zum Wert in der 3-SAT Instanz, da eine 3-Klausel erfüllt ist genau dann, wenn exakt zwei der drei 2-Klauseln erfüllt sind.

Daher sind die lokalen Optima in beiden Instanzen genau gleich. Die Voraussetzungen an eine PLS-Reduktion sind erfüllt, darüber hinaus bleiben sogar die Nachbarschaftsbeziehungen erhalten. □

**Satz 2.26.** *Pos-NAE-2-SAT  $\leq_{PLS}$  Max-Cut.*

*Beweis.* Jede Variable wird durch einen Knoten und jede Klausel mit ihrem Gewicht durch eine Kante repräsentiert. Mehrfachkanten werden in eine einzelne Kante zusammengefügt.

Bei einer Zuweisung für die Variablen interpretieren wir 0 als *Links* und 1 als *Rechts*. Auf diese Weise sind die lokalen Optima in beiden Instanzen genau gleich. Die Voraussetzungen an eine PLS-Reduktion sind erfüllt, darüber hinaus bleiben sogar die Nachbarschaftsbeziehungen erhalten. □

**Definition 2.27** (PLS-Vollständigkeit). Ein Problem  $\Pi^*$  in PLS heißt *PLS-vollständig*, wenn für jedes Problem  $\Pi$  in PLS gilt  $\Pi \leq_{\text{PLS}} \Pi^*$ .

Beispiele für PLS-vollständige Probleme:

- Eine Masterreduktion zeigt PLS-Vollständigkeit von FLIP.
- Eine PLS-Reduktion von FLIP zu Pos-NAE-3-SAT zeigt, dass das letztere Problem auch PLS-vollständig ist.
- Also sind Pos-NAE-2-SAT und Max-Cut ebenfalls PLS-vollständig.

**Satz 2.28** (Fabrikant, Papadimitriou, Talwar 2004). *Für die Komplexität von reinen Nash-Gleichgewichten in Auslastungsspielen gilt:*

|              | Netzwerkspiele           | allgemeine Auslastungsspiele |
|--------------|--------------------------|------------------------------|
| symmetrisch  | $\exists$ Poly-Zeit Algo | PLS-vollständig              |
| asymmetrisch | PLS-vollständig          | PLS-vollständig              |

Wir zeigen nur einen der PLS-Vollständigkeitsbeweise, den für allgemeine, asymmetrische Auslastungsspiele.

*Beweis.* Wir zeigen eine PLS-Reduktion von Max-Cut zu Auslastungsspielen.

Zuerst interpretieren wir Max-Cut als ein Spiel:

Spieler sind Knoten in einem gewichteten Graphen  $G = (V, E)$  mit je zwei Strategien: *Links* und *Rechts*. Ein Zustand des Spiels erzeugt einen *Schnitt*, d.h. eine Aufteilung von  $V$  in linke und rechte Knoten. Dabei erfassen die Kantengewichte die Antipathie zwischen den Spielern. Somit wählen die Spieler eine Strategie, so dass die Summe der Gewichte der inzidenten Kanten zur anderen Knotenmenge am höchsten ist.

Reine Nash-Gleichgewichte sind genau lokale Optima von Max-Cut.

Um Max-Cut in ein Auslastungsspiel zu transformieren, müssen Ressourcen definiert werden, über deren Latenzen die Spieler minimieren.

Zuerst interpretieren wir das definierte Max-Cut-Spiel als Minimierungsvariante:

Die Strategien eines Knotens sind weiterhin *Links* und *Rechts*. Allerdings sind die Kosten eines Spielers die Summe der Gewichte der inzidenten Kanten zu Spielern, welche die selbe Strategie spielen.

Beide Spiele haben den gleichen Übergangsgraphen:

Ein Spieler *minimiert* die Gewichte der inzidenten Kanten zu Spielern auf "seiner Seite"  $\Leftrightarrow$  Er *maximiert* die Gewichte der inzidenten Kanten zu Spielern auf der "anderen Seite".

Nun definieren wir die Ressourcen: Jede Kante  $e$  wird durch zwei Ressourcen  $e_{links}, e_{rechts}$  mit Latenzfunktionen  $d(1) = 0$  und  $d(2) = w_e$  repräsentiert. Für jeden Spieler enthält die Strategie  $S_{links}$  die Ressourcen  $e_{links}$  aller inzidenten Kanten; Strategie  $S_{rechts}$  enthält die Ressourcen  $e_{rechts}$  aller inzidenten Kanten.

Die Spieler in diesem Auslastungsspiel haben genau die gleichen Kosten wie die Spieler in der Minimierungsvariante des Max-Cut-Spiels.

Daher sind die reinen Nash-Gleichgewichte dieses Auslastungsspiels genau die lokalen Optima der Max-Cut Instanz. Damit haben wir eine PLS-Reduktion von Max-Cut zu Auslastungsspielen gezeigt.

□

## 2.2 Ordinale Potenziale und Schwache Kreisfreiheit

Ein exaktes Potenzial garantiert die endliche Verbesserungseigenschaft und damit die Existenz eines reinen Nash-Gleichgewichts. Man kann die Definition eines Potenzialspiels allerdings noch deutlich allgemeiner fassen, ohne dabei die endliche Verbesserungseigenschaft zu verlieren

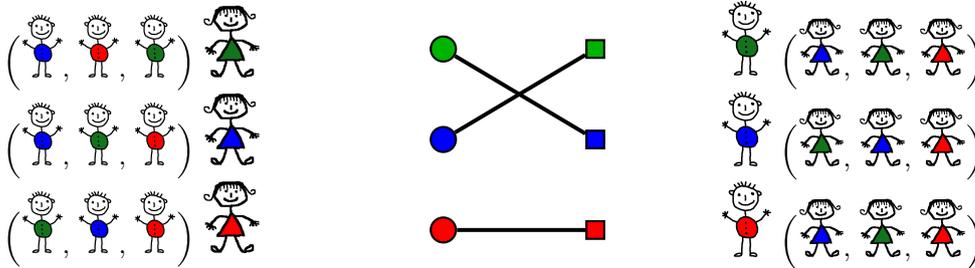
**Definition 2.29** (Ordinales Potenzialspiel). Ein strategisches Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$  besitzt eine *ordinale Potenzialfunktion*  $\Phi: \Sigma \rightarrow \mathbb{R}$ , wenn für jedes  $i \in \mathcal{N}$ , für jedes  $S_{-i} \in \Sigma_{-i}$ , und für jede  $S_i, S'_i \in \Sigma_i$  gilt:

$$c_i(S_i, S_{-i}) > c_i(S'_i, S_{-i}) \Rightarrow \Phi(S_i, S_{-i}) > \Phi(S'_i, S_{-i}) .$$

Ein solches Spiel nennen wir *ordinales Potenzialspiel*.

**Beobachtung:** Sei  $\Gamma$  ein ordinales Potenzialspiel. Dann hat  $\Gamma$  die *endliche Verbesserungseigenschaft*, und daher existiert ein reines Nash-Gleichgewicht.

**Beispiel 2.30.** *Partnerwahl*

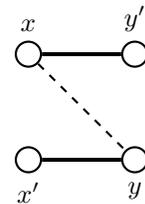


Die Frauen und Männer suchen jeweils einen Partner. Jeder Spieler besitzt eine Präferenzordnung über die Spieler der anderen Seite mit der Ordnung: (bester Partner, zweitbester, ...). Gesucht ist ein Matching, das den Präferenzen möglichst entspricht. ■

**Definition 2.31** (Matching mit Präferenzen). Es existiert eine Menge  $\mathcal{X}$  von  $n$  Männern und eine Menge  $\mathcal{Y}$  von  $n$  Frauen. Jeder Mann  $x \in \mathcal{X}$  hat eine *Präferenzordnung*  $\succ_x$  über alle  $y \in \mathcal{Y}$  (analog für die Frauen), dabei ist jeder lieber verpartnert als allein. Für ein Matching  $M$  sei  $M(x) \in \mathcal{Y}$  die Partnerin von Mann  $x \in \mathcal{X}$  in  $M$ , und  $M(y) \in \mathcal{X}$  der Partner von Frau  $y \in \mathcal{Y}$  in  $M$ . Sei  $M(x) = *$ , wenn  $x$  allein in  $M$ , und  $M(y) = *$  genauso.

Wann ist ein Matching stabil? Was ist eine Gefahr für Stabilität?

**Definition 2.32** (Stabiles Matching). In  $M$  ist  $\{x, y\}$  ein *blockierendes Paar* genau dann, wenn  $x$  und  $y$  sich gegenseitig lieber mögen als ihre jeweiligen Partner  $y' = M(x)$  und  $x' = M(y)$ . Ein Matching  $M$  ohne blockierendes Paar ist ein *stabiles Matching*.



Stabiles Matching spielt eine entscheidende Rolle in vielen Anwendungen wie Ärzte/Krankenhäuser-Zuweisung, Uni-Zulassung, Arbeitsmarkt, etc.

Wir betrachten das Modell hier als **strategisches Spiel**:

Die **Männer**  $\mathcal{X}$  sind die **Spieler** und die Strategiemenge für jeden Mann  $x \in \mathcal{X}$  ist die Menge der Frauen  $\Sigma_x = \mathcal{Y}$ . Als **Strategie** wählt ein Mann eine **Frau** und “macht ihr einen Antrag”.

Wir drücken die Präferenzordnungen durch Kosten aus: Jeder Match  $\{x, y\}$  liefert Kostenwerte  $c_x(y) > 0$  und  $c_y(x) > 0$  für die Beteiligten, wobei gilt

$$\begin{aligned} c_x(y) > c_x(y') &\Leftrightarrow y' \succ_x y , \\ c_y(x) > c_y(x') &\Leftrightarrow x' \succ_y x , \\ c_y(*) = c_x(*) = \infty &\text{ für alle } x \in \mathcal{X}, y \in \mathcal{Y}. \end{aligned}$$

In einem Zustand  $S$  wählt jeder Mann  $x \in \mathcal{X}$  eine Frau  $y_x \in \mathcal{Y}$ . In  $S$  erhält also jede Frau  $y \in \mathcal{Y}$  eine (evtl. leere) Menge  $A_y(S)$  von Anträgen. Ein Match ergibt sich nur mit  $x_y^* = \arg \min\{c_y(x) \mid x \in A_y(S)\}$ , also dem Mann der  $y$  aus der Menge  $A_y(S)$  am besten gefällt.  $x$  erhält in  $S$  also Kosten  $c_x(S) = c_x(M_S(x))$ , wobei  $M_S(x)$  sein Match in Zustand  $S$  ist (Beachte:  $M_S(x) = *$  ist möglich).

**Beobachtung:** Ein Zustand  $S$  im Spiel ist ein reines Nash-Gleichgewicht  
 $\Leftrightarrow$  Die Menge der Matches ist ein stabiles Matching.

**Gewichtetes stabiles Matching**

Wir betrachten hier zuerst **gewichtetes** oder **korreliertes stabiles Matching**. Jedes mögliche Paar  $e = \{x, y\}$  erzeugt dabei Kosten von  $c(x, y) > 0$  für beide Spieler. Es gilt also  $c_x(y) = c_y(x) = c(x, y)$ .

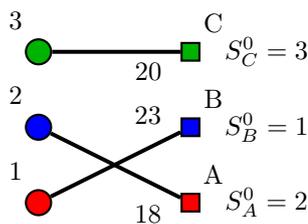
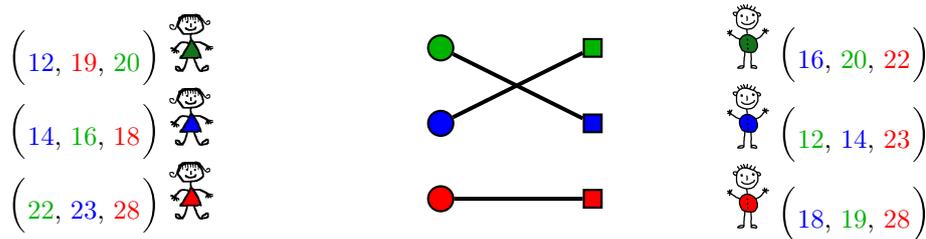
Die Präferenzen der Spieler sind über die Werte  $c(x, y)$  korreliert – je kleiner der Wert, desto attraktiver ist die Paarung für beide Partner.

**Satz 2.33.** *Gewichtetes stabiles Matching ist ein ordinales Potenzialspiel. Wenn alle Werte  $c(x, y)$  paarweise verschieden sind, dann ist das reine Nash-Gleichgewicht eindeutig.*

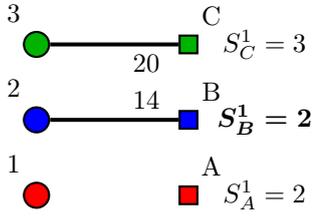
*Beweis.* Betrachte Zustand  $S$  und die Menge  $M_S$  der Matches. Wir nutzen als Potenzial  $\Phi(S)$  den Vektor der Kostenwerte der Männer  $c_x(S)$  sortiert in nicht-fallender Reihenfolge. Sei nun  $\Phi(S) < \Phi(S')$ , wenn der Vektor für Zustand  $S$  lexikografisch kleiner ist als der Vektor für Zustand  $S'$ .

Wir betrachten die folgende Beispiel-Instanz und durchlaufen eine Verbesserungsfolge vom Zustand  $S^0$ . Der sortierte Vektor der Kostenwerte der Spieler sinkt lexikografisch in jedem Verbesserungsschritt.

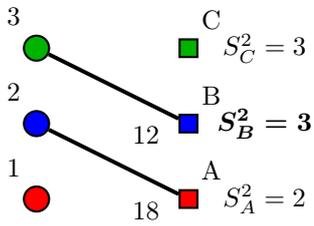
**Beispiel 2.34.** *Gewichtete Partnerwahl mit lexikographischer Sortierung*



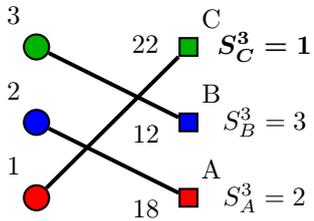
| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |



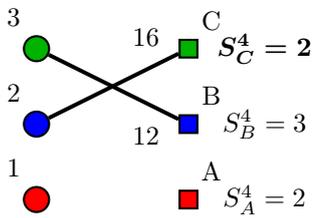
| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |
| $S^1$   | 14, 20, $\infty$ |



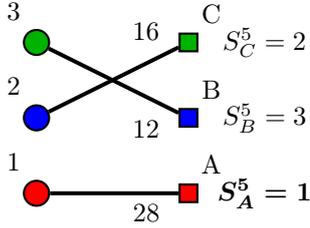
| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |
| $S^1$   | 14, 20, $\infty$ |
| $S^2$   | 12, 18, $\infty$ |



| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |
| $S^1$   | 14, 20, $\infty$ |
| $S^2$   | 12, 18, $\infty$ |
| $S^3$   | 12, 18, 22       |



| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |
| $S^1$   | 14, 20, $\infty$ |
| $S^2$   | 12, 18, $\infty$ |
| $S^3$   | 12, 18, 22       |
| $S^4$   | 12, 16, $\infty$ |



| Zustand | Sortierte Kosten |
|---------|------------------|
| $S^0$   | 18, 20, 23       |
| $S^1$   | 14, 20, $\infty$ |
| $S^2$   | 12, 18, $\infty$ |
| $S^3$   | 12, 18, 22       |
| $S^4$   | 12, 16, $\infty$ |
| $S^5$   | 12, 16, 28       |
|         | reines NG        |

■

Warum sinkt das lexikographische Potenzial immer, wenn sich ein Spieler strikt verbessert? Sei  $x$  ein Mann, der von Strategie  $y$  zu  $y'$  wechselt und sich dabei strikt verbessert. Seien  $S$  und  $S'$  die resultierenden Zustände. Da  $x$  sich strikt verbessert, ist er natürlich in  $S'$  zu  $y'$  gematcht. Je nachdem ob  $x$  und  $y'$  auch in  $S$  gematcht waren, ergeben sich vier Fälle (siehe unten).

In jedem Fall ist der kleinste hinzugefügte Kostenwert  $c(x, y')$ , und er ist immer strikt kleiner als der kleinste entfernte Kostenwert. Damit sinkt der sortierte Vektor der Kostenwerte der Männer lexikografisch ab. Das Spiel hat eine ordinale (genauer: eine lexikografische) Potenzialfunktion.

1.  $\{x, y\} \in M_S$ ,  $y'$  ungematcht in  $S$ :  
Es gilt  $c_x(S) = c(x, y) > c(x, y') = c_x(S')$ . In der sortierten Folge der Kostenwerte wird  $c(x, y)$  durch  $c(x, y')$  ersetzt. Wenn  $y$  noch einen weiteren Antrag in  $S$  außer von  $x$  hat, dann hat sie auch einen Antrag in  $S'$ . Dann wird für den besten Mann  $x'' \in A_y(S')$  in der sortierten Folge  $\infty$  durch  $c(x'', y)$  ersetzt.
2.  $\{x, y\} \notin M_S$ ,  $y'$  ungematcht in  $S$ :  
Es gilt  $c_x(S) = \infty > c(x, y') = c_x(S')$ . In der sortierten Folge der Kostenwerte wird  $\infty$  durch  $c(x, y')$  ersetzt.
3.  $\{x, y\} \in M_S$ ,  $\exists x' \in \mathcal{X}$  mit  $\{x', y'\} \in M_S$ :  
Es gilt  $c_x(S) = c(x, y) > c(x, y') = c_x(S')$ . Da  $\{x, y'\} \in M_{S'}$  gilt  $c(x', y') > c(x, y')$ . In der sortierten Folge der Kostenwerte wird  $c(x, y)$  durch  $c(x, y')$  und  $c(x', y')$  durch  $\infty$  ersetzt. Wenn  $y$  noch einen weiteren Antrag in  $S$  außer von  $x$  hat, dann hat sie auch einen Antrag in  $S'$ . Dann wird für den besten Mann  $x'' \in A_y(S')$  in der sortierten Folge  $\infty$  durch  $c(x'', y)$  ersetzt.
4.  $\{x, y\} \notin M_S$ ,  $\exists x' \in \mathcal{X}$  mit  $\{x', y'\} \in M_S$ :  
Es gilt  $c_x(S) = \infty > c(x, y') = c_x(S')$ . Da  $\{x, y'\} \in M_{S'}$  gilt  $c(x', y') > c(x, y')$ . In der sortierten Folge der Kostenwerte wird  $\infty$  durch  $c(x, y')$  und  $c(x', y')$  durch  $\infty$  ersetzt.

Seien nun alle Werte  $c(x, y)$  paarweise unterschiedlich. Betrachte das Paar  $\{x_0, y_0\}$  mit dem kleinsten Kostenwert. In jedem Zustand  $S$  ist  $y_0$  die eindeutige beste Antwort für Spieler  $x_0$ , denn der Match liefert die kleinsten Kosten und  $y_0$  wird seinen Antrag immer annehmen. Daher muss  $x_0$  in jedem reinen Nash-Gleichgewicht Strategie  $y_0$  wählen.

Von den übrigen möglichen Paaren betrachte nun wieder das verbleibende Paar  $\{x_1, y_1\}$  mit kleinstem Kostenwert (also mit  $x_1 \neq x_0$  und  $y_1 \neq y_0$ ). Unter der Bedingung, dass  $x_0$  und  $y_0$  beide vergeben sind, liefert der Match die kleinsten Kosten für  $x_1$  und  $y_1$ . Daher ist  $y_1$  die eindeutige beste Antwort für  $x_1$ , denn sie wird seinen Antrag annehmen. Daher muss  $x_1$  in jedem reinen Nash-Gleichgewicht Strategie  $y_1$  wählen.

Induktiv angewendet zeigt dieses Argument, dass das reine Nash-Gleichgewicht **eindeutig** ist. □

Mit der letzten Überlegung können wir auch Algorithmen konstruieren. Für eine kurze Verbesserungsfolge wählen wir in jedem Schritt den Spieler, der durch seine Abweichung den kleinsten Kostenwert erzielen kann. Dann weicht immer zuerst  $x_0$  zu  $y_0$  ab, dann  $x_1$  zu  $y_1$ , etc.

Der Greedy-Algorithmus wendet diese Strategie mit dem leeren Matching an und versucht, die Kanten in der Reihenfolge aufsteigender Kosten in das Matching einzufügen. Dann wird immer zuerst  $\{x_0, y_0\}$  eingefügt, dann  $\{x_1, y_1\}$ , etc.

**Korollar 2.35.**

1. Es gibt von jedem Zustand  $S$  eine Verbesserungsfolge zum reinen Nash-Gleichgewicht mit höchstens  $n$  Schritten.
2. Ein reines Nash-Gleichgewicht kann mit einem Greedy-Algorithmus in Zeit  $O(n^2 \log n)$  berechnet werden.

Das Korollar gilt auch, wenn die Werte  $c(x, y)$  nicht paarweise verschieden sind. (Warum?)

**Existenz und Berechnung**

Wir betrachten nun den allgemeinen Fall des Matchingspiels, in dem die Kosten eventuell nicht mehr korreliert sind. Der Einfachheit halber nehmen wir als *Kostenwert die Position des Partners in der Präferenzordnung*.

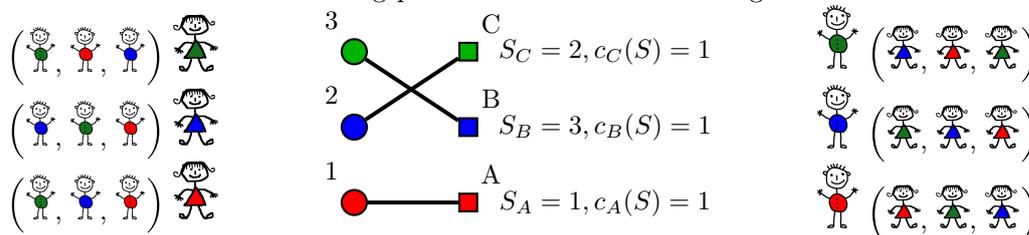
Sei  $\succ_x = (y_1, \dots, y_n)$  die Präferenzordnung von Mann  $x \in \mathcal{X}$ . Dann ist  $c_x(y_k) = k$  für  $k \in \{1, \dots, n\}$  und  $c_x(*) = n + 1$ . Die Kosten  $c_y(x)$  werden analog definiert. In einem Zustand  $S$  ergibt sich wieder ein Matching  $M_S$  und die Kosten  $c_x(S) = c_x(M_S(x))$ .

Wenn die Präferenzen der Spieler nicht korreliert sind, dann gibt es Matchingspiele, die keine endliche Verbesserungseigenschaft haben. Sogar wenn die Spieler nur **beste Antworten** spielen, kann es **kreisende Folgen** geben. Also gibt es i.A. keine ordinale Potenzialfunktion.

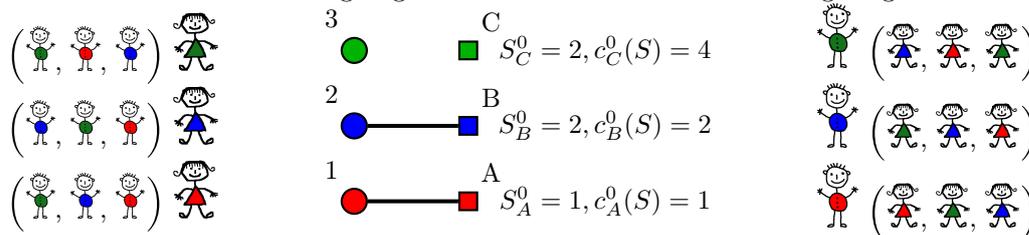
**Proposition 2.36.** *Es gibt Matchingspiele mit Folgen von besten Antworten, die in einen Kreis führen. Diese Spiele haben keine endliche Verbesserungseigenschaft.*

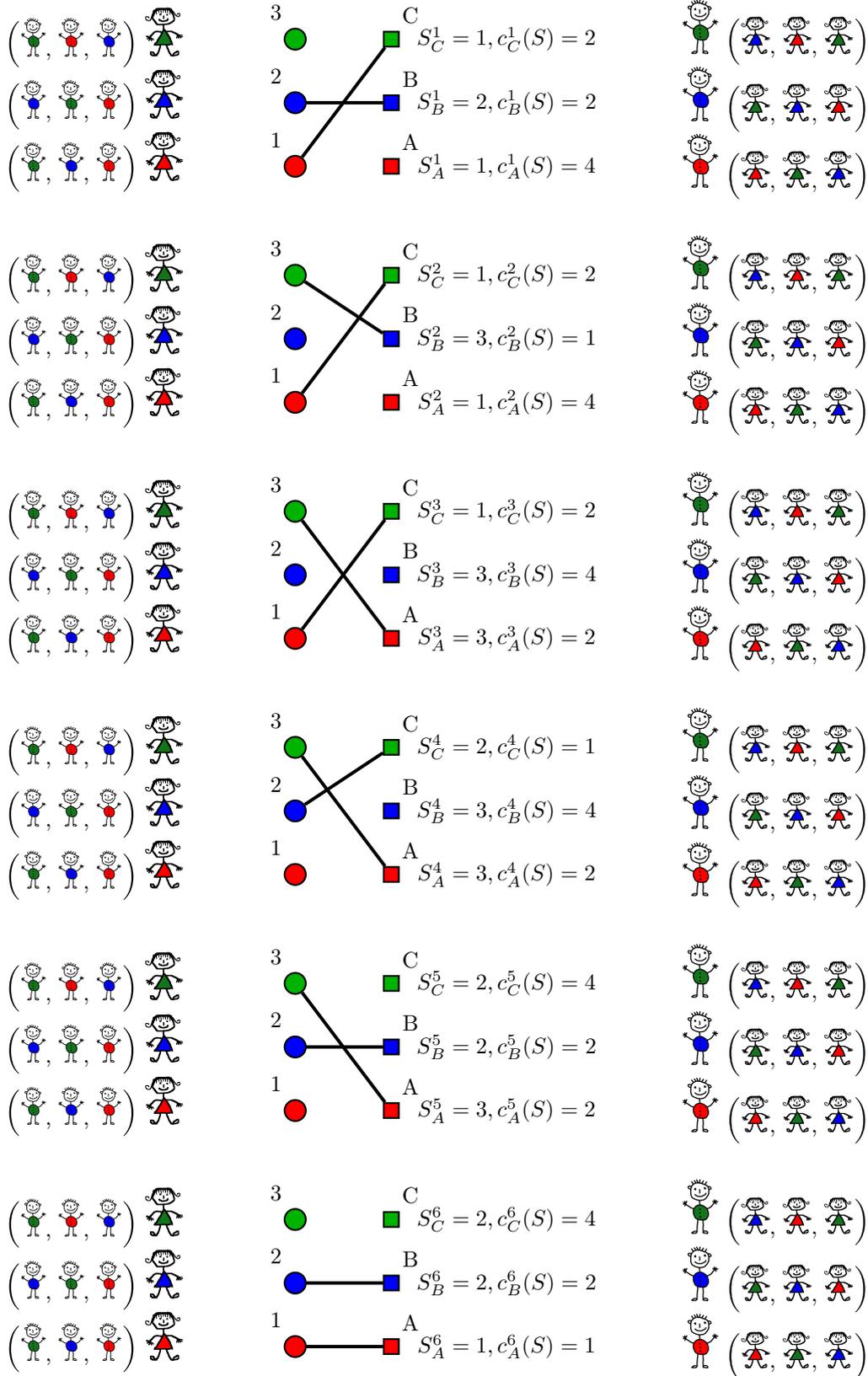
**Beispiel 2.37.** *Partnerwahl ohne endliche Verbesserungseigenschaft*

Wir betrachten ein Matchingspiel mit dem stabilen Matching:



Wir führen eine Verbesserungsfolge von besten Antworten durch und gelangen in einen Kreis.





Im Matchingspiel gibt es eventuell kreisende Verbesserungsfolgen, aber auch immer ein reines Nash-Gleichgewicht (stabiles Matching).

**Algorithmus 1:** Deferred-Acceptance (DA) Algorithmus mit Männerantrag

---

Initialisiere  $\succ'_x = \succ_x$  für alle  $x \in \mathcal{X}$   
**while** es gibt ungematchten Mann  $x \in \mathcal{X}$  mit  $\succ'_x \neq \emptyset$  **do**  
  Jeder Mann  $x \in \mathcal{X}$  wählt beste Frau in  $\succ'_x$  als Strategie  $S_x$   
  Jede Frau  $y \in \mathcal{Y}$  behält den besten Mann in  $A_y(S)$   
   $y$  lehnt jeden weiteren Antrag aus  $A_y(S)$  ab  
  Wenn sein Antrag abgelehnt wird, löscht Mann  $x$  die beste Frau aus  $\succ'_x$

---

**Satz 2.38** (Gale, Shapley 1962). *Jedes Matchingspiel hat ein stabiles Matching. Für gegebene Präferenzlisten kann ein stabiles Matching in Zeit  $O(n^2)$  berechnet werden.*

*Beweis.* Der DA Algorithmus berechnet ein Matching und kann offensichtlich in Zeit  $O(n^2)$  implementiert werden. Während der Ausführung des Algorithmus gilt:

1. Für jeden Mann sinkt seine Präferenz für die Frau, der er einen Antrag macht.
2. Für jede Frau steigt ihre Präferenz für den gematchten Partner.

Für einen Widerspruch nehmen wir an, das finale Matching  $M$  hat ein blockierendes Paar  $\{x, y\}$  mit  $y \succ_x M(x)$  und  $x \succ_y M(y)$ .  $x$  muss  $y$  einen Antrag gemacht haben und abgelehnt worden sein. Also hat  $y$  einen Antrag eines besseren Mannes bekommen  $x' \succ_y x$ . Daher kann sie nur zu einem (noch) besseren Mann in  $M$  gematched sein. Also gilt  $M(y) \succeq_y x' \succ_y x$ , ein Widerspruch zu  $x \succ_y M(y)$ . □

Mit einer Reformulierung dieser Idee zeigen wir eine noch stärkere Eigenschaft.

**Satz 2.39.** *Für jedes Matchingspiel und jeden initialen Zustand  $S^0$  gibt es eine Folge von höchstens  $2n^2$  besten Antworten in ein reines Nash-Gleichgewicht.*

*Beweis.* Die Folge hat zwei Phasen. In **Phase 1** dürfen nur **gematchte Männer** beste Antworten spielen. Sei  $X$  die Menge der gematchten Männer in  $S$ . Die Funktion  $\Phi$  sinkt in Phase 1:

$$\Phi(S) = \sum_{x \in X} c_x(S) + \sum_{x \in \mathcal{X} \setminus X} c_x(S_x). \quad (\text{Rang von } x\text{'s Partnerin in } \succ_x)$$

$\Phi(S)$  summiert für jeden gematchten Mann  $x$  den Rang seiner Partnerin und für jeden ungematchten Mann  $x$  den Rang der "virtuellen" Partnerin, d.h. den Rang der gewählten Frau  $S_x$  wenn er zu dieser Frau gematcht werden würde.

Wir betrachten nun den Fall, dass in einer Runde ein gematchter Mann  $x \in X$  von seiner Partnerin  $y = S_x$  zu einer besten Antwort  $y'$  abweicht. Hierbei machen wir drei Beobachtungen:

1.  $x$  bleibt gematched. Die Abweichung verbessert den Rang seiner Partnerin und verringert den Wert von  $\Phi$  um mind. 1.
2. Falls  $y'$  zu einem Mann  $x' \in X$  gematched war, dann wird  $x'$  ungematched und die Kosten  $c_{x'}(y')$  werden von der ersten Summe in die zweite Summe verschoben.
3. Falls  $y$  in  $S$  noch andere Anträge hat (d.h. es gibt mind. einen Mann  $x' \neq x$  mit  $S_{x'} = y$ ), dann bleibt  $y$  nach der Abweichung von  $x$  weiterhin gematched. Es gibt dann also einen ungematchten Mann  $x'' \in \mathcal{X} \setminus X$ , der nun zu  $y$  gematched wird. Die Kosten  $c_{x''}(y)$  werden von der zweiten Summe in die erste Summe verschoben.

Die letzten beiden Anpassungen verändern die Funktion  $\Phi$  nicht, die erste Anpassung bewirkt eine Verbesserung.  $\Phi$  sinkt also um mind. 1 in jeder Runde. Es gilt  $1 \leq \Phi(M) \leq n^2$ , also endet Phase 1 nach maximal  $n^2$  Runden.

In **Phase 2** dürfen nur **ungematchte Männer** beste Antworten spielen. Sei  $Y$  die Menge der gematchten Frauen in  $S$ . Die Funktion  $\Psi$  steigt in Phase 2:

$$\Psi(S) = \sum_{y \in Y} (n + 1 - c_y(S)).$$

In einer Runde weicht ein ungematchter Mann  $x$  zu einer besten Antwort ab. Entweder  $x$  wird gematched zu  $y \in Y$  und  $c_y(S)$  sinkt um mind. 1. oder  $x$  wird gematched zu  $y \notin Y$ , so dass  $y$  zu  $Y$  hinzugefügt wird.

$\Psi$  steigt also um mind. 1 in jeder Runde. Es gilt  $1 \leq \Psi(M) \leq n^2$ , also endet Phase 2 nach maximal  $n^2$  Runden.

Bleibt zu zeigen: Der finale Zustand ist ein stabiles Matching. In Phase 2 kann sich **kein gematchter Mann verbessern**. Wenn ein ungematchter Mann  $x$  zu Frau  $y$  gematched wird, verringert dies die Kosten von  $y$ . Wenn es vorher kein blockierendes Paar mit einem gematchten Mann gab, dann gibt es auch kein solches Paar nachdem  $x$  und  $y$  gematched sind –  $x$  hat eine beste Antwort gespielt und  $y$ 's Kosten sind gesunken. Am Ende von Phase 2 kann sich natürlich auch **kein ungematchter Mann verbessern**. □

Wir haben damit gezeigt, dass das Matchingspiel schwach kreisfrei ist

**Definition 2.40** (Schwach kreisfrei). Ein strategisches Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$  heißt *schwach kreisfrei* (engl.: weakly acyclic), wenn es für jeden Zustand  $S$  *mindestens eine endliche Verbesserungsfolge* gibt, die von  $S$  in ein reines Nash-Gleichgewicht führt.

Wir betrachten *zufällige Verbesserungsfolgen*: In jeder Runde wird uniform zufällig ein Spieler  $i \in \mathcal{N}$  ausgewählt. Er wählt uniform zufällig eine neue Strategie  $S'_i \in \Sigma_i$  und weicht ab, wenn sie eine strikte Verbesserung darstellt.

Eine zufällige Verbesserungsfolge ist ein Random Walk über die Zustände des Spiels. Reine Nash-Gleichgewichte sind absorbierende Zustände.

**Beobachtung:** In schwach kreisfreien Spielen konvergieren *zufällige Verbesserungsfolgen mit Wahrscheinlichkeit 1* im Grenzwert zum reinen Nash-Gleichgewicht. Wenn der Walk nur lange genug läuft, führt er irgendwann eine Verbesserungsfolge aus, die in ein reines Nash-Gleichgewicht führt.

Also konvergieren **zufällige Verbesserungsfolgen** in jedem Matchingspiel. Aber **wie lange dauert es** bis ein stabiles Matching erreicht ist? Was passiert, wenn wir zusätzlich auf **zufällige Folgen von besten Antworten** einschränken? Leider gibt es Spiele, in denen gilt mit hoher Wahrscheinlichkeit, dass die Konvergenzzeit exponentiell lang ist.

**Satz 2.41** (Ackermann, Goldberg, Mirrokni, Röglin, Vöcking 2011). *Es gibt ein Matchingspiel und einen Startzustand  $S^0$ , so dass mit Wahrscheinlichkeit  $1 - 2^{-\Omega(n)}$  gilt: Die zufällige Verbesserungsfolge von  $S^0$  braucht mindestens  $2^{\Omega(n)}$  Runden, um ein stabiles Matching zu erreichen. Dieses Resultat gilt auch für zufällige Folgen von besten Antworten.*

## 2.3 Zusammenfassung

Im Allgemeinen existieren reine Nash-Gleichgewichte nicht immer, jedoch in speziellen Spielen (z.B. Potenzialspielen), die die endliche Verbesserungseigenschaft besitzen. Die Berechnung eines reinen Nash-Gleichgewichts benötigt eine polynomielle Laufzeit in der Kostenmatrix (sehr groß).

Insbesondere ist die Berechnung PLS-schwer in kompakt repräsentierten Auslastungsspielen. Sogar die Beste-Antwort-Dynamik besitzt eine exponentielle und für spezielle Potenzialspele eine polynomielle Konvergenzzeit.

Zur Vertiefung wird folgende Literatur empfohlen:

- D. Monderer, L. Shapley. Potential Games. *Games and Economic Behavior* 14:1124–1143, 1996.
- A. Fabrikant, C. Papadimitriou, K. Talwar. The Complexity of Pure Nash Equilibria. *STOC* 2004.
- S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, Q. Sun. Fast and Compact: A Simple Class of Congestion Games. *AAAI* 2005. (Singletonspiele)
- H. Ackermann, H. Röglin, B. Vöcking. On the Impact of Combinatorial Structure on Congestion Games. *Journal of the ACM* 55(6), 2008. (Matroidspiele, PLS)
- S. Chien, A. Sinclair. Convergence to Approximate Nash Equilibria in Congestion Games. *Games and Economic Behavior* 71(2):315–327, 2011.
- A. Roth, J. Vande Vate. Random Paths to Stability in Two-Sided Matching. *Econometrica* 58(6):1475-1480, 1990. (Schwache Kreisfreiheit)
- D. Abraham, A. Levavi, D. Manlove, G. O'Malley. The Stable Roommates Problem with Globally Ranked Pairs. *Internet Math.* 5(4):493-515, 2008. (Ordinales Potenzial)
- F. Mathieu. Self-Stabilization in Preference-based Systems. *Peer-to-Peer Netw. Appl.* 1(2):104-121, 2008. (Ordinales Potenzial)
- H. Ackermann, P. Goldberg, V. Mirrokni, H. Röglin, B. Vöcking. Uncoordinated Two-Sided Matching Markets. *SIAM J. Comput.* 40(1):92-106, 2011.

## 3 | Gemischte Nash-Gleichgewichte

Im vorherigen Kapitel wurden reine Nash-Gleichgewichte thematisiert, bei denen kein Spieler unilateral von seiner reinen Strategie abweichen möchte. Im realen Leben gibt es oft nicht eine einzige Strategie, die jeder in Abhängigkeit der anderen Spieler wählen möchte. Ein berühmtes Beispiel für solch ein Problem ist das Spiel Schere-Stein-Papier. Dieses Spiel hat kein reines Nash-Gleichgewicht. Was können wir als Ergebnis in diesem Spiel erwarten?

In diesem Kapitel untersuchen wir ein universelles Lösungskonzept, ein Gleichgewicht mit garantierter Existenz in endlichen Spielen. Wir versuchen, das Gleichgewichtskonzept möglichst einfach zu halten, ohne große Einschränkungen oder Annahmen. Das gemischte Nash-Gleichgewicht wird sich als solches Konzept herausstellen. Neben der Existenzgarantie werden wir auch die Berechnung untersuchen. Lässt sich ein gemischtes Nash-Gleichgewicht effizient berechnen?

### Beispiel 3.1. Schere-Stein-Papier (Rock-Paper-Scissors)

Das bekannte Spiel Schere-Stein-Papier ist ein gutes Beispiel für Spiele ohne reine Nash-Gleichgewichte. Hier wählen beide Spieler zwischen den Strategien Schere, Stein und Papier, wobei jede Strategie gleich oft zum Gewinn bzw. Verlust führt. Des Weiteren kann ein Spieler nur gewinnen, wenn sein Gegner verliert. Angenommen die Spieler wählen jeweils eine reine Strategie, dann gibt es in jedem Zustand mindestens einen Spieler, der von seiner gewählten Strategie abweichen möchte.

Die vertikalen Pfeile im folgenden Schema verdeutlichen die präferierten Abweichungen des Zeilenspielers in den jeweiligen Zuständen; die horizontalen Pfeile signalisieren die präferierten Abweichungen des Spaltenspielers.

|   | R       | P       | S       |
|---|---------|---------|---------|
| R | 0<br>↓→ | -1<br>↓ | 1<br>←  |
| P | 1<br>→  | 0<br>↓→ | -1<br>↑ |
| S | -1<br>↑ | 1<br>←  | 0<br>←↑ |

Es ist deutlich zu erkennen, dass in jedem Zustand mindestens ein Spieler abweichen möchte, da er keine beste Antwort spielt. Es existiert also kein reines Nash-Gleichgewicht.

Bei diesem Spiel randomisieren die Spieler meist über ihre Strategien, damit der Gegner keine deterministische beste Antwort spielen kann. ■

**Definition 3.2** (Gemischte Strategie). Eine *gemischte Strategie*  $x_i$  für Spieler  $i$  ist eine Wahrscheinlichkeitsverteilung über die Menge der reinen Strategien  $S_i$ .

In endlichen Spielen gilt für  $x_i$ , dass  $x_{ij} \in [0, 1]$  und  $\sum_{j \in S_i} x_{ij} = 1$ .

Die Kosten eines gemischten Zustands für Spieler  $i$  sind

$$c_i(x) = \mathbb{E}_{s \sim x}[c_i(s)] = \sum_{s \in S} p(s) \cdot c_i(s) ,$$

wobei

$$p(s) = \prod_{i \in \mathcal{N}, j = s_i} x_{ij}$$

die Wahrscheinlichkeit ist, dass sich am Ende der (reine) Zustand  $s$  als Resultat ergibt.

**Definition 3.3** (Gemischte beste Antwort). Eine (*gemischte*) *beste Antwort*  $x_i$  auf eine Kollektion von gemischten Strategien  $x_{-i}$  erfüllt  $c_i(x_i, x_{-i}) \leq c_i(x'_i, x_{-i})$  für jede andere gemischte Strategie  $x'_i$ .

**Definition 3.4** (Gemischtes Nash-Gleichgewicht). Ein gemischter Zustand  $x$  heißt *Nash-Gleichgewicht in gemischten Strategien* (oder *gemischtes Nash-Gleichgewicht*), wenn  $x_i$  eine beste Antwort auf  $x_{-i}$  ist, für jeden Spieler  $1 \leq i \leq n$ .

**Bemerkung:** Jede reine Strategie ist eine gemischte Strategie. Jedes reine Nash-Gleichgewicht ist ein gemischtes Nash-Gleichgewicht.

**Beispiel 3.5.** *Spiel mit gemischtem Nash-Gleichgewicht*

Im folgenden Beispiel randomisieren die Spieler über ihre Strategien. Betrachte den folgenden gemischten Zustand:

|     |   |                             |                             |                             |
|-----|---|-----------------------------|-----------------------------|-----------------------------|
|     |   | 0.3                         | 0.7                         |                             |
|     |   | 2                           | 3                           | $0.3 \cdot 1 + 0.7 \cdot 2$ |
| 0.2 | 1 |                             |                             | $= 0.3 + 1.4$               |
|     |   |                             |                             | $= 1.7$                     |
|     |   | 4                           | 2                           | $0.3 \cdot 1 + 0.7 \cdot 5$ |
| 0.8 | 1 |                             |                             | $= 0.3 + 3.5$               |
|     |   |                             |                             | $= 3.8$                     |
|     |   | $0.2 \cdot 2 + 0.8 \cdot 4$ | $0.2 \cdot 3 + 0.8 \cdot 2$ |                             |
|     |   | $= 0.4 + 3.2$               | $= 0.6 + 1.6$               |                             |
|     |   | $= 3.6$                     | $= 2.2$                     |                             |

Durch die randomisierten Strategien lassen sich die entstehenden Kosten nicht mehr direkt aus dem Diagramm ablesen. Angenommen der Zeilenspieler wählt seine erste Strategie, der er mit Wahrscheinlichkeit 0.2 folgt. Mit Wahrscheinlichkeit 0.3 wählt der Spaltenspieler seine erste Strategie, wodurch der Zeilenspieler die Kosten 1 erhält. Mit Wahrscheinlichkeit 0.7 entstehen analog die Kosten 2. Daraus ergeben sich die erwarteten Kosten  $0.3 \cdot 1 + 0.7 \cdot 2 = 1.7$  für den Zeilenspieler, wenn er seine erste Strategie wählt. Die gegebenen gemischten Strategien bilden kein gemischtes Nash-Gleichgewicht. Der Zeilenspieler erhält die erwarteten Kosten  $c_1(x) = 0.2 \cdot 1.7 + 0.8 \cdot 3.8 > 1.7$ , daher kann er seine Kosten durch Abweichen zur reinen Strategie (1,0) minimieren. Analog erhält der Spaltenspieler die erwarteten Kosten  $c_2(x) = 0.3 \cdot 3.6 + 0.7 \cdot 2.2 > 2.2$  und kann seine Kosten durch Abweichen zur reinen Strategie (0,1) minimieren. ■

Im obigen Beispiel ist  $x$  kein gemischtes Nash-Gleichgewicht, weil mindestens ein Spieler mit positiver Wahrscheinlichkeit eine suboptimale Strategie auswählt.

**Fakt 3.6.** Wenn  $x_{ij} > 0$  in einer gemischten besten Antwort  $x_i$  auf  $x_{-i}$ , dann ist  $j$  eine reine beste Antwort auf  $x_{-i}$ .

Die Kosten von  $x_i$  sind ein *gewichteter Durchschnitt* von Kosten für die reinen Strategien. Dieser gewichtete Durchschnitt ist minimal, genau dann wenn der Durchschnitt nur über reine Strategien mit minimalen Kosten gebildet wird.

**Beispiel 3.7.** *Spiel mit gemischtem Nash-Gleichgewicht*

Wie oben erläutert ist der Zustand  $x$  mit  $x_1 = (1, 0)$  und  $x_2 = (1, 0)$  ein reines (und somit auch gemischtes) Nash-Gleichgewicht.

|   | 1                                | 0                                |                                  |
|---|----------------------------------|----------------------------------|----------------------------------|
| 1 | 2                                | 3                                | $1 \cdot 1 + 0 \cdot 2$<br>$= 1$ |
| 0 | 4                                | 2                                | $1 \cdot 1 + 0 \cdot 5$<br>$= 1$ |
|   | $1 \cdot 2 + 0 \cdot 4$<br>$= 2$ | $1 \cdot 3 + 0 \cdot 2$<br>$= 3$ |                                  |

Für den Zeilenspieler sind bei dem Verhalten des Gegners beide Strategien optimal. Daher kann der Zeilenspieler über beide Strategien randomisieren, ohne seine Kosten zu erhöhen.

|               | 1  | 0  |                                  |
|---------------|--|--|----------------------------------|
| $\frac{2}{3}$ | 2  | 3  | $1 \cdot 1 + 0 \cdot 2$<br>$= 1$ |
| $\frac{1}{3}$ | 4  | 2  | $1 \cdot 1 + 0 \cdot 5$<br>$= 1$ |
|               | $\frac{2}{3} \cdot 2 + \frac{1}{3} \cdot 4$<br>$= \frac{8}{3}$ | $\frac{2}{3} \cdot 3 + \frac{1}{3} \cdot 2$<br>$= \frac{8}{3}$ |                                  |

Der Zustand  $x$  mit  $x_1 = (\frac{2}{3}, \frac{1}{3})$  und  $x_2 = (1, 0)$  ist ein gemischtes Nash-Gleichgewicht. Für den Reihenspieler ist die obere Strategie eine dominante Strategie, aber in der ersten Spalte ist sie nicht *strikt* besser. Wenn sie in jeder Spalte strikt besser wäre, dann würde die untere Strategie in keinem gemischten Nash-Gleichgewicht gespielt. (Warum?)

■

### 3.1 Existenz von gemischten Nash-Gleichgewichten

Der zentrale Satz dieses Kapitels ist der Satz von Nash.

**Satz 3.8** (Satz von Nash). *Jedes endliche strategische Spiel hat mindestens ein gemischtes Nash-Gleichgewicht.*

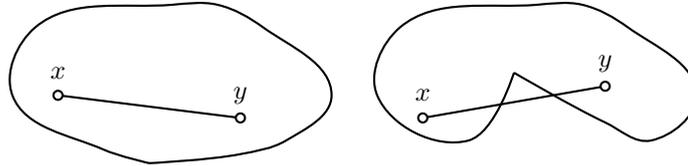
*Beweis.* Wir nutzen den Fixpunktsatz von Brouwer zum Beweis.

**Satz 3.9** (Fixpunktsatz von Brouwer). *Jede stetige Funktion  $f : D \rightarrow D$ , die eine kompakte und konvexe Menge  $D \subseteq \mathbb{R}^m$  auf sich selbst abbildet, hat mindestens einen Fixpunkt  $x^* \in D$  mit  $f(x^*) = x^*$ .*

Wir definieren die Voraussetzungen für den Fixpunktsatz:

- Eine Menge  $D \subset \mathbb{R}^m$  ist *konvex* genau dann, wenn für jedes  $x, y \in D$  und jedes  $\lambda \in [0, 1]$  gilt  $\lambda x + (1 - \lambda)y \in D$ .

Die Graphik stellt links eine konvexe Menge dar. Die rechte Menge besitzt diese Eigenschaft nicht.

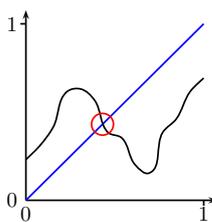


- Eine Menge  $D \subset \mathbb{R}^m$  ist *kompakt* genau dann, wenn sie abgeschlossen und beschränkt ist.
- Eine Menge  $D \subseteq \mathbb{R}^m$  ist *beschränkt* genau dann, wenn es eine ganze Zahl  $M \geq 0$  gibt mit  $D \subseteq [-M, M]^m$ .
- Betrachte eine Menge  $D \subseteq \mathbb{R}^m$  und eine Folge  $x_0, x_1, \dots$ , so dass für alle  $i \geq 0$  gilt  $x_i \in D$  und  $x \in \mathbb{R}^m$  existiert mit  $x = \lim_{i \rightarrow \infty} x_i$  (d.h., für alle  $\epsilon > 0$  gibt es eine ganze Zahl  $k > 0$  mit  $\|x - x_j\|_2 < \epsilon$  für alle  $j > k$ ). Eine Menge  $D$  ist *abgeschlossen*, wenn  $x \in D$  für jede solche Folge.
- Eine Funktion  $f : D \rightarrow \mathbb{R}^m$  ist stetig am Punkt  $x \in D$ , wenn für alle  $\epsilon > 0$  ein  $\delta > 0$  existiert, so dass für alle  $y \in D$  gilt: Wenn  $\|x - y\|_2 < \delta$ , dann  $\|f(x) - f(y)\|_2 < \epsilon$ .  $f$  heißt *stetig*, wenn sie an jedem Punkt  $x \in D$  stetig ist.

Die folgenden Beispiele verdeutlichen die Definitionen von *abgeschlossen* und *beschränkt*:

- $[0, 1]^2$  ist abgeschlossen und beschränkt.
- $[0, 1)$  ist nicht abgeschlossen (und nicht offen), aber beschränkt.
- $[0, \infty)$  ist abgeschlossen und unbeschränkt.

**Beispiel 3.10.** Funktion über dem Einheitsintervall



Jede stetige Funktion  $f : [0, 1] \rightarrow [0, 1]$  hat einen Fixpunkt:

Die blaue Gerade entspricht allen möglichen Fixpunkten. Angenommen für die Funktion  $f$  gilt  $f(0) > 0$  und  $f(1) < 1$  (sonst gäbe es einen Fixpunkt bei 0 oder 1), dann muss  $f$  aufgrund der Stetigkeit die blaue Gerade mindestens einmal kreuzen. An jeder dieser Kreuzungen befindet sich ein Fixpunkt. ■

Für den Beweis des Satzes von Nash prüfen wir zuerst die Bedingungen für den Fixpunktsatz.

**Fakt 3.11.** Die Menge  $X$  der gemischten Zustände  $x = (x_1, \dots, x_n)$  in einem endlichen strategischen Spiel ist eine konvexe, kompakte Teilmenge von  $\mathbb{R}^m$  mit  $m = \sum_{i=1}^n m_i$  und  $m_i = |S_i|$ .

Wir definieren eine stetige Funktion  $f : X \rightarrow X$ , die jeden gegebenen Zustand in einen anderen Zustand transformiert. Die Fixpunkte von  $f$  werden genau die gemischten Nash-Gleichgewichte des Spiels sein.

Ein gemischtes Nash-Gleichgewicht  $x$  ist eine Kollektion von gemischten besten Antworten  $x_i$ . Wenn  $x_{ij} > 0$  in einer besten Antwort  $x_i$  auf  $x_{-i}$ , dann ist  $j \in S_i$  eine reine beste Antwort auf  $x_{-i}$ .

Eine Kollektion von besten Antworten (also ein gemischtes Nash-Gleichgewicht)  $x = (x_1, \dots, x_n)$  erfüllt

$$c_i(x) - c_i(j, x_{-i}) \leq 0 \quad \text{für jedes } j \in S_i \text{ und jedes } i \in \mathcal{N}.$$

Was ist eine passende Definition von  $f$ ? Die Funktion soll einen Fixpunkt  $x$  besitzen genau dann, wenn  $x$  ein gemischtes Nash-Gleichgewicht ist.

Für einen gemischten Zustand  $x$  sei

$$\phi_{ij}(x) = \max\{0, c_i(x) - c_i(j, x_{-i})\} .$$

$\phi_{ij}(x)$  gibt an, ob der Spieler sich mit Hilfe der Strategie  $j$  verbessern kann. In diesem Fall liefert  $j$  geringere Kosten als die momentane Strategie  $x_i$  und  $\phi_{ij}(x)$  liefert einen positiven Wert. Ist  $\phi_{ij}(x) = 0$ , verringert  $j$  die Kosten nicht.

Sei  $f : X \rightarrow X$  mit  $f(x) = x' = (x'_1, \dots, x'_n)$  gegeben durch

$$x'_{ij} = \frac{x_{ij} + \phi_{ij}(x)}{1 + \sum_{k=1}^{m_i} \phi_{ik}(x)}$$

für jedes  $i = 1, \dots, n$  und  $j = 1, \dots, m_i$ .

Intuitiv verteilt  $f$  die Wahrscheinlichkeiten um, so dass die Strategien mit geringen Kosten mit höherer Wahrscheinlichkeit gewählt werden. Gleichzeitig werden die Wahrscheinlichkeiten normiert, damit diese sich immer auf 1 summieren.

**Fakt 3.12.**  $f$  erfüllt die Voraussetzungen von Brouwers Fixpunktsatz:  $f$  ist stetig und wenn  $x \in X$ , dann ist  $f(x) = x' \in X$  ein gemischter Zustand.

**Beispiel 3.13.** Anwendung der Funktion  $\phi$

Der Spieler  $i$  hat 3 reine Strategien zur Auswahl, wobei die momentane gemischte Strategie  $x_i = (0.2, 0.5, 0.3)$  lautet und die Kosten  $c(x_i, x_{-i}) = 3.2$  produziert. Mit den Strategiewahlen  $x_{-i}$  ergeben sich die Kosten der Strategien  $c_i(\cdot, x_{-i}) = (2.2, 4.2, 2.2)$ . Unter diesen Umständen wird Strategie  $x_i$  abgebildet auf  $x'_i$ :

| $x_{ij}$ | $c_i(j, x_{-i})$ | $\phi_{ij}(x)$ | $x'_{ij}$                         |
|----------|------------------|----------------|-----------------------------------|
| 0.2      | 2.2              | 1              | $\frac{0.2+1}{1+2} = 0.4$         |
| 0.5      | 4.2              | 0              | $\frac{0.5+0}{1+2} \approx 0.166$ |
| 0.3      | 2.2              | 1              | $\frac{0.3+1}{1+2} \approx 0.434$ |

■

Mit Brouwers Satz wissen wir, dass ein  $x^*$  mit  $f(x^*) = x^*$  existiert. Wir müssen zwei Richtungen zeigen:

$$f(x) = x \Leftrightarrow x \text{ ist gemischtes Nash-Gleichgewicht} .$$

“ $\Rightarrow$ ”: Sei  $x$  ein gemischtes Nash-Gleichgewicht, dann spielen alle Spieler ausschließlich optimale Strategien mit positiver Wahrscheinlichkeit. Dadurch kann sich kein Spieler durch Abweichen zu einer anderen Strategie  $j$  verbessern, so dass für alle  $\phi_{ij}(x) = 0$ . Daraus resultiert  $x'_{ij} = \frac{x_{ij}}{1}$ .

“ $\Leftarrow$ ”: Für jedes  $i = 1, \dots, n$  und  $j = 1, \dots, m_i$  gilt

$$x^*_{ij} = \frac{x^*_{ij} + \phi_{ij}(x^*)}{1 + \sum_{k=1}^{m_i} \phi_{ik}(x^*)} ,$$

daher

$$x^*_{ij} \cdot \left( 1 + \sum_{k=1}^{m_i} \phi_{ik}(x^*) \right) = x^*_{ij} + \phi_{ij}(x^*) ,$$

und

$$x_{ij}^* \sum_{k=1}^{m_i} \phi_{ik}(x^*) = \phi_{ij}(x^*) .$$

Wir werden zeigen, dass  $\sum_{k=1}^{m_i} \phi_{ik}(x^*) = 0$ . Das heißt, dass in  $x_i^*$  nur reine beste Antworten positive Wahrscheinlichkeit erhalten, also ist  $x_i^*$  eine gemischte beste Antwort.

**Behauptung 3.14.** *Für jeden gemischten Zustand  $x$  und jeden Spieler  $i \in \mathcal{N}$  gibt es eine reine Strategie  $j \in S_i$  mit  $x_{ij} > 0$  und  $\phi_{ij}(x) = 0$ .*

*Beweis (Behauptung 3.14).*  $c_i(x) = \sum_{j=1}^{m_i} x_{ij} \cdot c_i(j, x_{-i})$ , daher muss es ein  $j$  geben mit  $x_{ij} > 0$  und Kosten mindestens so hoch wie das gewichtete Mittel.

Formal gibt es ein  $j$  mit  $x_{ij} > 0$  und

$$c_i(x) - c_i(j, x_{-i}) \leq 0 .$$

Daher gilt  $\phi_{ij}(x) = \max\{0, c_i(x) - c_i(j, x_{-i})\} = 0$ . □ [Behauptung]

Für jeden Spieler  $i$  betrachten wir Strategie  $j$  aus der Behauptung. Das bedeutet  $x_{ij}^* > 0$  und

$$x_{ij}^* \cdot \sum_{k=1}^{m_i} \phi_{ik}(x^*) = \phi_{ij}(x^*) = 0 .$$

Da  $x_{ij}^* > 0$ , ergibt sich

$$\sum_{k=1}^{m_i} \phi_{ik}(x^*) = 0 ,$$

und daher  $\phi_{ik}(x^*) = 0$  für alle  $k = 1, \dots, m_i$ . Daraus folgt

$$c_i(x^*) \leq c_i(j, x_{-i}^*) \quad \text{für alle } j \in S_i .$$

Das bedeutet, dass  $x_i^*$  eine beste Antwort ist, und der Satz von Nash ist gezeigt. □ [Satz]

## 3.2 Komplexität

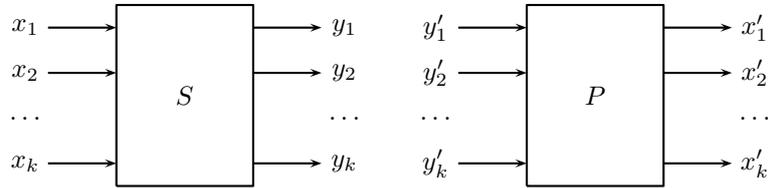
Wir wissen, dass jedes endliche strategische Spiel mindestens ein gemischtes Nash-Gleichgewicht besitzt. Nun beschäftigen wir uns mit der Frage, ob man diese Gleichgewichte effizient berechnen kann. Was ist die Komplexität des Problems?

Diese Problemstellung ist anders als die, die wir normalerweise betrachten. Das Problem entspricht keiner Optimierung und keiner nicht-trivialen Entscheidung (da Existenz garantiert). Stattdessen stellt sich uns das Suchproblem, ein Nash-Gleichgewicht zu *finden*. Wir betrachten die Komplexitätsklasse PPAD (polynomial parity argument, directed case) und ein Konzept der PPAD-Vollständigkeit (ähnlich wie bei NP). Dazu definieren wir ein PPAD-vollständiges Problem und konstruieren Reduktionen in Polynomialzeit.

Es gibt Spiele mit 3 Spielern und rationalen Kostenwerten, in denen alle gemischten Nash-Gleichgewichte irrationale Einträge haben. Daher können wir numerisch immer nur *Approximationen von gemischten Nash-Gleichgewichten* oder Fixpunkten berechnen.

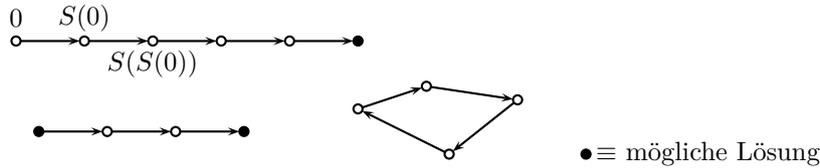
### 3.2.1 END-OF-LINE

END-OF-LINE ist ein PPAD-vollständiges Problem. Eine Instanz von END-OF-LINE ist ein Suchproblem, welches aus den beiden Schaltkreisen  $S$  und  $P$  besteht.



Beide Schaltkreise  $S$  (successor) und  $P$  (predecessor) besitzen die gleiche Anzahl an Input- und Output-Bits. Zusammen definieren  $S$  und  $P$  einen gerichteten Graphen. Jeder Knoten dieses Graphen entspricht genau einem  $k$ -Bit-Vektor. Die gerichtete Kante  $(x, y)$  existiert, wenn  $S(x) = y$  und  $P(y) = x$ . Daneben gibt es genau zwei zusätzliche Annahmen: Der 0-Vektor hat (1) keine eingehende Kante und (2) mindestens eine ausgehende Kante.

Die Problemstellung entspricht dem Finden eines Quellknotens (keine eingehende Kante) oder Senkeknotens (keine ausgehende Kante), der nicht der 0-Vektor ist. Ein solcher Graph kann zum Beispiel so aussehen:



**Beobachtung:** Jeder Knoten hat den Eingangsgrad und Ausgangsgrad  $\leq 1$ , daher besitzt END-OF-LINE durch das Paritätsargument immer eine Lösung. Gesucht wird nicht notwendigerweise das Ende des Pfades, der bei 0 startet. Diese spezielle Senke zu finden ist PSPACE-vollständig.

Die Eingabe einer END-OF-LINE Instanz ist nur durch die Schaltkreise  $S$  und  $P$  gegeben. Der Graph ist exponentiell groß in der Eingabe, kann also nicht in Polynomialzeit explizit aufgebaut und durchsucht werden.

Eine Lösung für END-OF-LINE zu finden ist PPAD-vollständig. Es wird vermutet, dass kein effizienter Algorithmus für dieses Problem existiert.

### 3.2.2 Finden von (approximativen) Brouwer-Fixpunkten

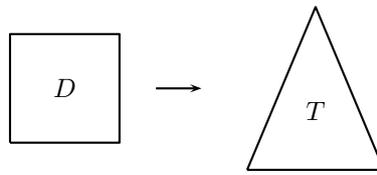
**Lemma 3.15.** *Das Finden eines (approximativen) gemischten Nash-Gleichgewichts ist in PPAD.*

*Beweisskizze.* Wir werden eine Reduktion verwenden und Brouwer-Fixpunkte mit END-OF-LINE finden. Dafür wird eine Aufteilung des Raumes in eine endliche Anzahl kleinerer Bereiche, die den Bit-Vektoren entsprechen, verwendet. Zu finden ist ein Bereich, der dem Fixpunkt ähnelt (Approximation). Durch die Stetigkeit liefert eine feinere Aufteilung eine präzisere Approximation.

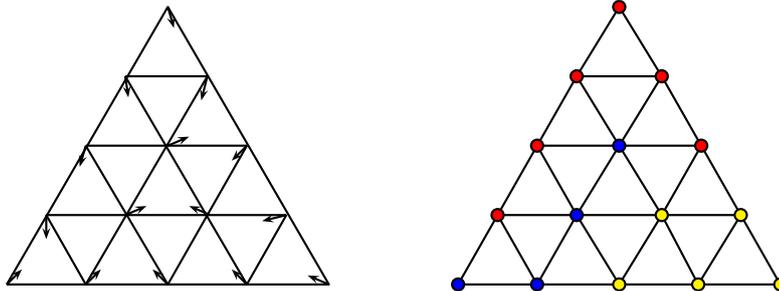
Teile den Raum in Simplexes (“multidimensionale Dreiecke”) und färbe Knoten aufgrund der Richtung der Fixpunkt-Funktion.

Wir beschränken uns hier zur einfachen Darstellung auf Probleme mit  $D \subseteq \mathbb{R}^2$ , z.B.,  $f : [0, 1]^2 \rightarrow [0, 1]^2$ , zur besseren Veranschaulichung. Dadurch ist die folgende Skizze eher als Beispiel anstatt eines allgemeinen Beweises zu interpretieren.

Der Einfachheit halber transformieren wir die Representation von  $[0, 1]^2$  in ein Dreieck  $T$  und erhalten ein äquivalentes Fixpunktproblem mit  $f' : T \rightarrow T$ .



Das Dreieck  $T$  wird unterteilt in kleinere Dreiecke. Für jeden Eckpunkt betrachte seine Abbildung unter  $f'$  und die Richtung, in der diese Abbildung liegt (linkes Dreieck). Abhängig von der Richtung erhält der Eckpunkt eine Farbe. Mit steigender Granularität werden die dreifarbigen Dreiecke die Fixpunkte von  $f'$ .



**Definition 3.16** (Unterteiltes Dreieck). Ein *unterteiltes Dreieck* ist eine Unterteilung eines Dreiecks in kleinere Dreiecke.

**Definition 3.17** (Sperner-Färbung). Eine *Sperner-Färbung* der Eckpunkte eines unterteilten Dreiecks erfüllt:

- Jeder extreme Eckpunkt erhält eine andere Farbe.
- Ein Eckpunkt an einer Außenseite des großen Dreiecks erhält eine Farbe der zwei Eckpunkte der Außenseite.
- Innere Eckpunkte können beliebig gefärbt sein.

Beachte: Die Färbung basierend auf den Richtungen der Fixpunktfunktion  $f'$  ist eine Sperner-Färbung.

**Lemma 3.18** (Sperners Lemma). *Jede Sperner-Färbung eines unterteilten Dreiecks hat ein dreifarbiges Dreieck.*

*Beweis.* Verbinde alle Eckpunkte auf der blau-rotten Außenseite zum blauen extremalen Eckpunkt. Starte auf der Außenseite und bewege Dich über Linien, die einen blauen und einen roten Punkt verbinden. Es gibt maximal 2 dieser Linien in jedem kleinen Dreieck, wir besuchen kein Dreieck doppelt.

Die Konstruktion ergibt eine Instanz von END-OF-LINE, in dem die kleinen Dreiecke die Knoten symbolisieren. Des Weiteren gibt es eine gerichtete Kante, wenn zwei kleine Dreiecke eine gemeinsame Linie zwischen einem roten und einem blauen Eckpunkt haben. Durch die Konstruktion besitzt jeder Knoten einen Eingangs- und Ausgangsgrad  $\leq 1$ . Zusätzlich gibt es durch die Konstruktion eine Quelle, andere Quellen/Senken sind dreifarbiges Dreiecke.



### 3.3.1 2-Spieler Nullsummenspiele

Ein Nullsummenspiel mit 2 Spielern wird als Matrix  $A \in \mathbb{Z}^{k \times \ell}$  mit  $k = |S_I|$  Reihen und  $\ell = |S_{II}|$  Spalten repräsentiert:

$$\begin{pmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1\ell} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2\ell} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{k1} & a_{k2} & \cdot & \cdot & \cdot & a_{k\ell} \end{pmatrix}$$

Dabei ist  $a_{ij}$  der Nutzenwert für Spieler I und  $-a_{ij}$  der Nutzenwert für Spieler II.

#### Beispiel 3.22. Nullsummenspiele und Nutzenberechnung

Hier einige Spiele in Matrixdarstellung mit Nutzenwerten.

|  |  |  |
|--|--|--|
| Matching Pennies                                 | Rock-Paper-Scissors  | Ein Spiel mit $k \neq \ell$ :                          |
| $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ | $\begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$ | $\begin{pmatrix} 0 & 2 & 4 \\ 1 & 2 & 3 \end{pmatrix}$ |

Im letzten Spiel mit  $k \neq \ell$  illustrieren wir die Berechnung des Nutzens. Wir bezeichnen gemischte Strategien mit  $x$  für I und  $y$  für II. Berechnung des Nutzens  $u_I(x, y)$ :

$$\begin{array}{c|ccc} & 0.1 & 0.4 & 0.5 \\ \hline 0.8 & 0 & 2 & 4 \\ 0.2 & 1 & 2 & 3 \end{array} \longrightarrow \begin{aligned} & 0.8 \cdot (0.1 \cdot 0 + 0.4 \cdot 2 + 0.5 \cdot 4) \\ & + 0.2 \cdot (0.1 \cdot 1 + 0.4 \cdot 2 + 0.5 \cdot 3) \\ & = 2.48 \end{aligned}$$

■

Allgemein gilt:

$$\begin{aligned} u_I(x, y) &= -u_{II}(x, y) = \sum_{i=1}^k \sum_{j=1}^{\ell} x_i a_{ij} y_j \\ &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \\ &= x^T A y . \end{aligned}$$

### 3.3.2 Öffentliche Strategiewahl

Wir nehmen an, I muss sich zuerst entscheiden. Er wählt eine gemischte Strategie, die II sehen kann *bevor* er seine eigene Wahl trifft. Wie sollte I diese öffentliche Strategie wählen?

#### Beispiel 3.23. Offene Strategiewahl in einem Nullsummenspiel

$$\begin{pmatrix} 0 & 2 & 4 \\ 1 & 2 & 3 \end{pmatrix}$$

II wird I so stark wie möglich verletzen, um seinen eigenen Nutzen zu maximieren. In diesem Spiel wird II immer mit Spalte 1 antworten. Daher ist die optimale Wahl für I die reine Strategie 2 (bzw:  $x = (0, 1)^T$ ). ■

I wählt  $x$ , dann wählt II  $y$  als beste Antwort auf  $x$ . II löst also das Problem  $\max_y u_{II}(x, y)$ , oder in Matrixform  $\max_y -x^T Ay = \min_y x^T Ay$ . Folglich sucht I nach einer Strategie  $x$ , die  $\min_y x^T Ay$  maximiert.

**Definition 3.24** (Gain-Floor). Der *Gain-Floor* eines Nullsummenspiels mit 2 Spielern ist der Wert

$$v_I^* = \max_x \min_y x^T Ay .$$

Eine Strategie  $x^*$ , die den Gain-Floor als Nutzen liefert, ist eine *optimale Strategie* (oder auch *Maximin-Strategie*) für I.

**Beispiel 3.25.** *Maximin*

Betrachte folgendes Spiel.

$$\begin{pmatrix} 5 & 1 & 2 \\ 1 & 4 & 3 \end{pmatrix}$$

II wird I so stark wie möglich verletzen.

- I wählt Reihe 1  $\Rightarrow$  II wählt Spalte 2  $\Rightarrow$  I hat Nutzen 1
  - I wählt Reihe 2  $\Rightarrow$  II wählt Spalte 1  $\Rightarrow$  I hat Nutzen 1
  - I wählt  $x = (0.5, 0.5)$ , minimaler Verlust für II ist 2.5 in Spalten 2 und 3
- $\Rightarrow$  I hat Nutzen 2.5.

Wie sieht die optimale Strategie  $x^*$  aus, wie groß kann  $v_I^*$  werden? ■

Intuitiv ist Spieler I benachteiligt, weil er seine Strategie zuerst veröffentlichen muss. Um seinen Nutzen abzuschätzen, betrachten wir den komplementären Fall, wenn Spieler II seine Strategie zuerst veröffentlichen muss. Nehmen wir nun an, II wählt zuerst  $y$ , dann wählt I  $x$  optimal mit  $\max_x u_I(x, y) = \max_x x^T Ay$ . Folglich sucht II nach  $y$ , um  $\max_x x^T Ay$  zu minimieren.

**Definition 3.26** (Loss-Ceiling). Die *Loss-Ceiling* eines Nullsummenspiels mit 2 Spielern ist der Wert

$$v_{II}^* = \min_y \max_x x^T Ay .$$

Eine Strategie  $y^*$ , die die Loss-Ceiling als Nutzen liefert, ist eine *optimale Strategie* (oder auch *Minimax-Strategie*) für II.

Wie sieht  $y^*$  aus, wie klein kann  $v_{II}^*$  werden? Welche Beziehung haben  $v_I^*$  und  $v_{II}^*$ ?

Wenn beide Spieler optimal spielen, dann sollte I mindestens  $v_I^*$  erhalten und II nicht mehr als  $v_{II}^*$  verlieren. Es ist einfach einzusehen, dass

**Lemma 3.27.** *Es gilt  $v_I^* \leq v_{II}^*$ .*

Überraschenderweise haben von Neumann and Morgenstern gezeigt:

**Satz 3.28** (Minimax Theorem). *In jedem Nullsummenspiel mit 2 Spielern gilt  $v = v_I^* = v_{II}^*$ . Der Wert  $v$  wird Wert des Spiels genannt.*

*Beweis.* Betrachte das Optimierungsproblem, bei dem wir  $x^*$  und  $v_I^* = \max_x \min_y x^T A y$  finden sollen. II wählt eine beste Antwort  $y$  auf  $x$ , um seinen Nutzen zu maximieren. Daher wählt II  $y_j > 0$  für ein gegebenes  $x$  nur dann, wenn in Spalte  $j$  der erwartete Verlust  $\sum_{i=1}^k x_i a_{ij}$  minimal ist. Daher gilt

$$v_I = \sum_{j=1}^{\ell} \sum_{i=1}^k x_i a_{ij} y_j = \min_{j=1, \dots, \ell} \sum_{i=1}^k x_i a_{ij} .$$

Für jedes  $x$  und den resultierenden Nutzen  $v_I$  von I wissen wir daher

$$v_I \leq \sum_{i=1}^k x_i a_{ij} \quad \text{für alle } j = 1, \dots, \ell.$$

Durch diese Bedingungen lässt sich das Optimierungsproblem für I als Lineares Programm darstellen:

$$\begin{aligned} & \text{Maximiere} && v_I \\ & \text{so dass} && v_I - \sum_{i=1}^k x_i a_{ij} \leq 0 && \text{für alle } j = 1, \dots, \ell \\ & && \sum_{i=1}^k x_i = 1 && (3.1) \\ & && x_i \geq 0 && \text{für alle } i = 1, \dots, k \\ & && v_I \in \mathbb{R} \end{aligned}$$

Die gleichen Argumente ergeben ein lineares Programm für die Minimierung der Loss-Ceiling.

$$\begin{aligned} & \text{Minimiere} && v_{II} \\ & \text{so dass} && v_{II} - \sum_{j=1}^{\ell} a_{ij} y_j \geq 0 && \text{für alle } i = 1, \dots, k \\ & && \sum_{j=1}^{\ell} y_j = 1 && (3.2) \\ & && y_j \geq 0 && \text{für alle } j = 1, \dots, \ell \\ & && v_{II} \in \mathbb{R} \end{aligned}$$

In Kapitel 3.3.3 wird gezeigt, dass dieses lineare Programm dual ist zum LP (3.1) für Gain-Floor-Maximierung.

Das Finden optimaler Strategien für I und II kann durch duale lineare Programme formuliert werden. Intuitiv wird die Maximierung von  $v_I$  gelöst, indem wir eine obere Schranke für die gegebenen Einschränkungen ermitteln. Die obere Schranke wird sich als  $v_{II}$  ergeben. Analog lässt sich  $v_{II}$  nach unten durch  $v_I$  beschränken.

Die *starke Dualität* in der Linearen Optimierung ist eine sehr grundlegende Eigenschaft. Betrachte ein lineares Programm mit einer gültigen Optimallösung, sei  $f^*$  der optimale Wert der Zielfunktion. Dann hat das duale Programm eine gültige Optimallösung mit Zielfunktionswert  $g^*$ . Die starke Dualität besagt, dass  $f^* = g^*$  gilt.

Starke Dualität liefert das Minimax-Theorem. □

**Beispiel 3.29.** *Aufstellung von dualen linearen Programmen*

Betrachte nochmals das Spiel aus Beispiel 3.25:

$$\begin{pmatrix} 5 & 1 & 2 \\ 1 & 4 & 3 \end{pmatrix}$$

|   |  |
|---|--|
| <p>Max. <math>v_I</math></p> <p>s.t. <math>v_I - 5x_1 - 1x_2 \leq 0</math></p> <p style="padding-left: 2em;"><math>v_I - 1x_1 - 4x_2 \leq 0</math></p> <p style="padding-left: 2em;"><math>v_I - 2x_1 - 3x_2 \leq 0</math></p> <p style="padding-left: 4em;"><math>x_1 + x_2 = 1</math></p> <p style="padding-left: 4em;"><math>x_1, x_2 \geq 0</math></p> <p style="padding-left: 4em;"><math>v_I \in \mathbb{R}</math></p> <p style="padding-left: 2em;"><math>x^* = (0.4, 0.6)</math></p> <p style="padding-left: 2em;"><math>v_I^* = 2.6</math></p> | <p>Min. <math>v_{II}</math></p> <p>s.t. <math>v_{II} - 5y_1 - 1y_2 - 2y_3 \geq 0</math></p> <p style="padding-left: 2em;"><math>v_{II} - 1y_1 - 4y_2 - 3y_3 \geq 0</math></p> <p style="padding-left: 4em;"><math>y_1 + y_2 + y_3 = 1</math></p> <p style="padding-left: 4em;"><math>y_1, y_2, y_3 \geq 0</math></p> <p style="padding-left: 4em;"><math>v_{II} \in \mathbb{R}</math></p> <p style="padding-left: 2em;"><math>y^* = (0.2, 0, 0.8)</math></p> <p style="padding-left: 2em;"><math>v_{II}^* = 2.6</math></p> |
|---|--|

Ist  $(x^*, y^*)$  ein gemischtes Nash-Gleichgewicht? ■

**Korollar 3.30.**

*$(x, y)$  im Nullsummenspiel mit 2 Spielern ist gemischtes Nash-Gleichgewicht*

$\Leftrightarrow$   *$x$  und  $y$  sind optimale Strategien.*

*Beweis. " $\Rightarrow$ ":*

Betrachte  $(x, y)$  und sei  $x$  suboptimal. Dann gibt es ein  $y'$  mit  $u_{II}(x, y') > -v$ , und daher  $u_I(x, y') < v$ . Für ein gemischtes Nash-Gleichgewicht muss gelten  $u_{II}(x, y) \geq u_{II}(x, y')$ , daraus folgt  $u_I(x, y) < v$ . Wenn  $u_I(x, y) < v$ , dann hat I mit einer optimalen Strategie einen höheren Nutzen. Wenn  $x$  suboptimal ist, liegt kein gemischtes Nash-Gleichgewicht vor. (Beweis ähnlich für  $y$  suboptimal)

*" $\Leftarrow$ ":*

Seien beide Strategien optimal, aber I hat eine bessere beste Antwort  $x'$ . Das bedeutet  $u_I(x', y) > v$ , aber dann ist  $y$  eine suboptimale Strategie für II. (Analoges Argument, wenn II eine bessere Strategie hat.) □

**Korollar 3.31.** *Jedes gemischte Nash-Gleichgewicht in einem Nullsummenspiel mit 2 Spielern ergibt den gleichen erwarteten Nutzen von  $v$  ( $-v$ ) für I (II).*

Es gibt effiziente Algorithmen zur Lösung von linearen Programmen. Wenn wir damit die linearen Programme (3.1) und (3.2) lösen, erhalten wir optimale Strategien. Damit ergibt sich folgendes Resultat:

**Satz 3.32.** *In jedem Nullsummenspiel mit 2 Spielern kann ein gemischtes Nash-Gleichgewicht in polynomieller Zeit berechnet werden.*

### 3.3.3 Appendix: LP-Dualität

Wir leiten eine obere Schranke auf  $v_I$  her für jede Lösung von (3.1).

Sei  $(v_I, x)$  eine Lösung von (3.1). Wir nutzen eine lineare Kombination der Constraints für die obere Schranke. Mit Faktoren  $z_j$  und  $w_I$  ergibt sich:

$$\begin{aligned} z_j \cdot \left( v_I - \sum_{i=1}^k x_i a_{ij} \right) &\leq z_j \cdot 0 \quad \text{für jedes } j \text{ und} \\ w_I \cdot \sum_{i=1}^k x_i &= w_I \cdot 1 \end{aligned}$$

Wir nehmen  $z_j \geq 0$ , um die richtige Richtung der Ungleichung einzuhalten.

Jetzt leiten wir obere Schranke her mit einer Linearkombination:

$$\begin{aligned} v_{\text{I}} &\leq \sum_{j=1}^{\ell} z_j \left( v_{\text{I}} - \sum_{i=1}^k x_i a_{ij} \right) + w_{\text{I}} \cdot \sum_{i=1}^k x_i \\ &= \left( \sum_{j=1}^{\ell} z_j \right) \cdot v_{\text{I}} + \sum_{i=1}^k \left( w_{\text{I}} - \sum_{j=1}^{\ell} a_{ij} z_j \right) \cdot x_i \\ &\leq \sum_{j=1}^{\ell} z_j \cdot 0 + w_{\text{I}} \cdot 1 = w_{\text{I}} \end{aligned}$$

Das klappt nur, wenn die erste Ungleichung erfüllt ist, und das ist der Fall, wenn die folgenden Bedingungen für die Koeffizienten der  $v_{\text{I}}$  und  $x_i$  auf der linken und rechten Seite gelten:

$$\begin{aligned} 1 &= \sum_{j=1}^{\ell} z_j && \text{(Gleich da } v_{\text{I}} \in \mathbb{R}.) \\ 0 &\leq w_{\text{I}} - \sum_{j=1}^{\ell} a_{ij} z_j && \text{(Evtl größer da } x_i \geq 0.) \end{aligned}$$

Was ist die beste obere Schranke  $w_{\text{I}}$ , die wir so erreichen können?

$$\begin{aligned} \text{Minimiere} \quad & w_{\text{I}} \\ \text{so dass} \quad & w_{\text{I}} - \sum_{j=1}^{\ell} a_{ij} z_j \geq 0 && \text{für alle } i = 1, \dots, k \\ & \sum_{j=1}^{\ell} z_j = 1 && (3.3) \\ & z_j \geq 0 && \text{für alle } j = 1, \dots, \ell \\ & w_{\text{I}} \in \mathbb{R} \end{aligned}$$

Dieses lineare Programm wird das *duale Programm* von (3.1) genannt,  $w_{\text{I}}$  und  $z_j$  sind die *dualen Variablen*. Dies repräsentiert genau das Optimierungsproblem (3.2), die Loss-Ceiling und eine optimale Strategie für II zu finden (mit  $w_{\text{I}} = v_{\text{II}}$  und  $z_j = y_j$ )! Für mehr Hintergrund zu Linearer Optimierung, Dualität, und Algorithmen wird die Lektüre "Introduction to Algorithms" von Cormen, Leiserson, Rivest, Stein empfohlen.

### 3.4 Zusammenfassung

Mit dem Satz von Nash wurde die Existenz gemischter Nash-Gleichgewichte in endlichen Spielen gezeigt. Die Berechnung eines approximativen gemischten Nash-Gleichgewichts ist laut Spernerschem Lemma in PPAD und wurde aufgrund von grundlegenden Resultaten in Literatur als PPAD-vollständig klassifiziert. Für 2-Spieler-Nullsummenspiele ist ein Gleichgewicht in polynomialer Zeit mit Hilfe von linearer Programmierung berechenbar.

Zur Vertiefung wird folgende Literatur empfohlen:

- G. Owen. Game Theory. Academic Press, 2001. (Kapitel 1 + 2)
- Nisan et al. Algorithmic Game Theory. Cambridge University Press, 2007. (Kapitel 1 + 2)
- J. Nash. Non-cooperative Games. Annals of Mathematics 54, pp. 286–295, 1951.
- P. Goldberg, C. Daskalakis, C. Papadimitriou. The Complexity of Computing a Nash Equilibrium. SIAM Journal on Computing, 39(1), pp. 195–259, 2009.
- X. Chen, X. Deng, S.-H. Teng. Settling the Complexity of Computing Two-Player Nash Equilibria. Journal of the ACM, 56(3), 2009.
- Hintergrund zu Linearer Optimierung, Dualität, und Algorithmen: Cormen, Leiserson, Rivest, Stein. Introduction to Algorithms, 3rd edition. MIT Press, 2009. (Kapitel 29)

# 4 | Korrelierte und grob-korrelierte Gleichgewichte

In den meisten Fällen müssen Entscheidungen mehrmals getroffen werden, so dass man auf die gemachten Erfahrungen reagieren kann. Ist es möglich ein Gleichgewicht durch Lernen zu erreichen? Welche Gleichgewichte werden in solch einer Situation erreicht?

Dazu betrachten wir in diesem Kapitel zwei Verallgemeinerungen von Nash-Gleichgewichten. Wir fordern die Existenz der Gleichgewichte, wobei sie zusätzlich "schnell" ermittelbar sein sollen.

### Beispiel 4.1. Ampel-Spiel

Zwei Autofahrer rasen auf eine Kreuzung zu und wollen möglichst ohne zu Bremsen und ohne Unfall weiterfahren.

|           | (G)as | (B)remsen |
|-----------|-------|-----------|
| (G)as     | 101   | 2         |
| (B)remsen | 0     | 1         |

Mit den gegebenen Kosten stellen die Zustände (G,B) und (B,G) reine Nash-Gleichgewichte dar. Des Weiteren existiert zum Beispiel das gemischte (nicht reine) Nash-Gleichgewicht mit den gemischten Strategien  $x_G = 0.01$  und  $x_B = 0.99$  für beide Spieler.

Nun erweitern wir das Spiel mit einer Ampel an der Kreuzung, welche den Spielern eine Strategie vorgibt. Jeweils die Hälfte der Zeit zeigt sie beidem Spielern "Gas" bzw. "Bremsen" an. Sollen sich die Spieler nach der Ampel richten, oder minimieren sie ihre Kosten durch Abweichen?

|           | (G)as | (B)remsen |
|-----------|-------|-----------|
| (G)as     | 101   | 2         |
| (B)remsen | 0     | 1         |

Angenommen die Ampel signalisiert einem Spieler zu bremsen, dann wird der Gegner Gas geben. In dem Fall will der Spieler einen Unfall vermeiden und bremst. Wenn ein Spieler laut Ampel Gas geben soll, bremst der Gegner, so dass der Spieler wieder nicht abweichen möchte. Es weicht also kein Spieler unilateral vom vorgegebenen Zustand ab (korreliertes Gleichgewicht).

Da beide Strategien jeweils mit positiver Wahrscheinlichkeit gespielt werden, aber nicht jeder Zustand erreicht wird, liegt kein gemischtes Nash-Gleichgewicht vor. ■

Im Folgenden werden zwei Konzepte für Gleichgewichte betrachtet, bei denen bestimmte Zustände mit gegebener Wahrscheinlichkeit ausgewürfelt werden. Unter den beschriebenen Bedingungen will jeweils kein Spieler von den gewürfelten Zuständen abweichen.

**Definition 4.2** (Korreliertes Gleichgewicht). Sei  $\mathcal{V}$  eine Wahrscheinlichkeitsverteilung über die Zustände eines endlichen Spiels.  $\mathcal{V}$  heißt *korreliertes Gleichgewicht*, wenn für jeden Spieler  $i \in \mathcal{N}$  und jede Austauschfunktion  $\sigma : S_i \rightarrow S_i$  gilt

$$\mathbb{E}_{s \sim \mathcal{V}}[c_i(s)] \leq \mathbb{E}_{s \sim \mathcal{V}}[c_i(\sigma(s_i), s_{-i})] .$$

$\mathcal{V}$  heißt (*additives*)  $\varepsilon$ -*approximatives korreliertes Gleichgewicht*, wenn

$$\mathbb{E}_{s \sim \mathcal{V}}[c_i(s)] \leq \mathbb{E}_{s \sim \mathcal{V}}[c_i(\sigma(s_i), s_{-i})] + \varepsilon .$$

**Definition 4.3** (Grob-korreliertes Gleichgewicht). Sei  $\mathcal{V}$  eine Wahrscheinlichkeitsverteilung über die Zustände eines endlichen Spiels.  $\mathcal{V}$  heißt *grob-korreliertes Gleichgewicht*, wenn für jeden Spieler  $i \in \mathcal{N}$  und jede Strategie  $s'_i \in S_i$  gilt

$$\mathbb{E}_{s \sim \mathcal{V}}[c_i(s)] \leq \mathbb{E}_{s \sim \mathcal{V}}[c_i(s'_i, s_{-i})] .$$

$\mathcal{V}$  heißt (*additives*)  $\varepsilon$ -*approximatives grob-korreliertes Gleichgewicht*, wenn

$$\mathbb{E}_{s \sim \mathcal{V}}[c_i(s)] \leq \mathbb{E}_{s \sim \mathcal{V}}[c_i(s'_i, s_{-i})] + \varepsilon .$$

Im gemischten Gleichgewicht wählt jeder Spieler eine *eigene Verteilung*  $x_i$  über seine *Strategien*. Die Verteilung über Zustände ergibt sich erst aus der *unabhängigen Kombination* der  $x_i$ . Keiner der Spieler möchte zu einer anderen (reinen) Strategie abweichen.

In korrelierten und grob-korrelierten Gleichgewichten wird *direkt eine Verteilung über Zustände* angegeben, die die Strategiewahlen einzelner Spieler miteinander korrelieren kann (z.B. als Resultat von gemeinsamem Lernen über die Zeit).

Keiner der Spieler möchte zu einer anderen (reinen) Strategie abweichen:

- **Korreliert:** *Unter der Bedingung, dass für Spieler  $i$  Strategie  $s_i$  ausgewürfelt wurde*, möchte  $i$  nicht zu einer anderen reinen Strategie abweichen.  $i$  erfährt dabei die für ihn ausgewürfelte Strategie. Die für die anderen Spieler ausgewürfelten Strategien erfährt  $i$  dagegen nicht.
- **Grob-korreliert:** Keiner der Spieler möchte *immer* zu einer reinen Strategie abweichen (hier wird nicht auf eine ausgewürfelte Strategie  $s_i$  bedingt). Spieler  $i$  erfährt hier nichts über die ausgewürfelten Strategien.

**Beispiel 4.4.** *Unterschied von korrelierten und grob-korrelierten Gleichgewichten*

Dieses Beispiel soll eine Intuition für die obigen Definitionen vermitteln und die Unterschiede verdeutlichen.

Dafür betrachten wir zuerst ein Spiel mit zwei Spielern und drei Strategien, bei dem sechs Zustände mit gleicher Wahrscheinlichkeit gewürfelt werden. Ist ein korreliertes/grob-korreliertes Gleichgewicht gegeben?

|   | D                        | E                        | F                              |
|---|--------------------------|--------------------------|--------------------------------|
| A | $\frac{1}{6}$<br>3<br>-4 | $\frac{1}{6}$<br>-3<br>4 | 90<br>90                       |
| B | 100<br>100               | 3<br>3<br>-4             | -3<br>-3<br>$\frac{1}{6}$<br>4 |
| C | $\frac{1}{6}$<br>4       | 110<br>110               | 3<br>3<br>$\frac{1}{6}$<br>-4  |

Angenommen es ist ein korreliertes Gleichgewicht gegeben, dann will kein Spieler von einem gegebenen Zustand zu einer anderen Strategie abweichen. Wir betrachten den Zeilenspieler, wenn ihm "A" vorgegeben wird. Er erfährt die Strategie des Spaltenspielers nicht, nur über die Verteilung kann er sehen, dass für den Gegner "D" bzw. "E" mit jeweils 50% ausgewürfelt wird. Der Zeilenspieler kann nun zu "B" oder "C" abweichen, wobei er jeweils in der Hälfte der Fälle hohe Kosten produziert, weswegen der Spieler nicht abweichen möchte. Wenn dem Zeilenspieler "B" bzw. "C" vorgeschlagen wird, gibt es analog keine Strategie mit niedrigeren erwarteten Kosten, also möchte der Spieler von keinem Zustand abweichen. Die Argumentation gilt analog für den Spaltenspieler, weswegen ein korreliertes und damit auch ein grob-korreliertes Gleichgewicht gegeben ist.

Wir betrachten das gleiche Spiel mit einer veränderten Wahrscheinlichkeitsverteilung. Ist hier ein korreliertes/grob-korreliertes Gleichgewicht gegeben?

|   | D          | E          | F        |
|---|------------|------------|----------|
| A | 3<br>-4    | 0.5<br>4   | 90<br>90 |
| B | 100<br>100 | 3<br>-4    | -3<br>4  |
| C | -3<br>4    | 110<br>110 | 3<br>-4  |

Wieder betrachten wir den Zeilenspieler, wenn ihm die Strategie "A" vorgegeben wird. In diesem Fall kann er seine Kosten durch ein Abweichen zu Strategie "B" minimieren, da durch die Verteilung nun klar ist, dass sein Gegner die Strategie "E" spielen wird. Analog will er von "B" zu "A" und von "C" zu "A" abweichen. Es liegt also kein korreliertes Gleichgewicht vor.

Angenommen es ist ein grob-korreliertes Gleichgewicht gegeben, dann gibt es keine Strategie, zu der ein Spieler immer abweichen möchte. Wenn der Zeilenspieler immer zu "A" abweichen würde, hätte er in einem viertel der Fälle die Kosten 90. Ebenso würde er seine Kosten erhöhen, wenn er immer "B" oder "C" wählt. Da sich dieses Argument analog auf den Spaltenspieler übertragen lässt, liegt ein grob-korreliertes Gleichgewicht vor. ■

## 4.1 Lernen von grob-korrelierten Gleichgewichten

### 4.1.1 Imitation von Experten

In vielen Fällen trifft man Entscheidungen nicht einmal sondern wiederholt, z.B. jeden Tag, ohne zu wissen wie sich andere Teilnehmer/Spieler oder die "Natur" an diesem Tag verhalten werden. Dazu betrachten wir Lernalgorithmen, die mit dieser Art von Unsicherheit umgehen können.

#### Beispiel 4.5. Weg zur Uni

Du musst jeden Morgen zur Uni fahren, dafür gibt es mehrere Verkehrsmöglichkeiten wie Auto, Fahrrad, Bus oder Tram. Du kannst nicht vorhersehen, ob es regnen wird und du daher nicht mit dem Rad fahren möchtest, oder ob z.B. Stau die Fahrt mit dem Auto behindert. Zu diesen  $n$  Verkehrsmöglichkeiten existieren  $n$  Experten, die jeden Morgen eine Vorhersage für das beste (geringste Kosten) Verkehrsmittel machen. Von diesen Experten wählst du jeden Morgen einen aus, um dessen Rat zu befolgen. Erst während der Fahrt stellt sich heraus, welches Verkehrsmittel an diesem Tag die beste Wahl gewesen wäre.

Wir werden zeigen, dass du auf lange Sicht im Durchschnitt *fast so gut sein kannst wie der beste Experte*, nur indem du die Wahlen der Experten imitierst. ■

Wir betrachten ein Gegnermodell mit diskreten Zeitschritten  $1, \dots, T$ . Sei  $[T] = \{1, \dots, T\}$ .

- Es gibt  $N$  *Experten*, nummeriert von 1 bis  $N$ .
- Im Schritt  $t \in [T]$  erfährt Experte  $i \in [N]$  die *Kosten*  $\ell_i^t \in [0, 1]$ , die von einem unbekanntem Gegner gewählt werden (z.B. der "Natur").
- Sei  $L_i^t = \sum_{k=1}^t \ell_i^k$  der Gesamtwert der Kosten von Experte  $i$  bis Zeitpunkt  $t$ .

Wenn die Experten imitiert werden, wählt ein *Online-Algorithmus*  $H$  in jedem Schritt einen Experten in Abhängigkeit der bisherigen Kosten.

- In Schritt  $t$  wählt  $H$  den Experten  $i \in [N]$  mit Wahrscheinlichkeit  $p_i^t$ .
- Der Vektor  $p^t$  kann von den vorherigen Kosten  $\ell^1, \dots, \ell^{t-1}$  abhängen.
- Der (erwartete) Verlust von  $H$  in Schritt  $t$  ist  $\ell_H^t = \sum_{i \in [N]} p_i^t \ell_i^t$ .
- Sei  $L_H^t = \sum_{k=1}^t \ell_H^k$  die (erwarteten) Gesamtkosten des Algorithmus bis Zeitschritt  $t$ .

#### 4.1.2 No-Regret-Algorithmen

Wir betrachten zuerst einen Greedy-Algorithmus, der auf die bisherigen Kosten reagiert. Im Folgenden sei  $L_{min}^{t-1} = \min_{i \in [N]} L_i^{t-1}$  für  $1 \leq t \leq T$  die Kosten des bisher besten Experten.

---

##### Algorithmus 2: Greedy-Algorithmus

---

**for** jeden Schritt  $t$  **do**

- Sei  $S^{t-1} = \{i : L_i^{t-1} = L_{min}^{t-1}\}$  die Menge der Experten  $i$  mit minimalen Kosten
  - Sei  $j = \min\{S^{t-1}\}$  der lexikographisch kleinste Experte mit minimalen Kosten
  - Setze  $p_j^t = 1$ , und  $p_i^t = 0$ , für  $i \neq j$
- 

Für die Analyse des Greedy-Algorithmus nehmen wir der Einfachheit halber an, dass alle Verluste entweder 0 oder 1 sind (anstatt reelle Zahlen in  $[0, 1]$ ).  
Wie gut ist der Algorithmus?

**Beispiel 4.6.** Eine worst-case Instanz für den Greedy-Algorithmus

|                        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |
|------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|
| $\ell_1$               | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1  | 0  | 0  | 1  | 0  |
| $L_1$                  | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4  | 4  | 4  | 5  | 5  |
| $\ell_2$               | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0  | 1  | 0  | 0  | 1  |
| $L_2$                  | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3  | 4  | 4  | 4  | 5  |
| $\ell_3$               | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0  | 0  | 1  | 0  | 0  |
| $L_3$                  | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3  | 3  | 4  | 4  | 4  |
| $j$                    | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1  | 2  | 3  | 1  | 2  |
| $\ell_{\text{Greedy}}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  |
| $L_G$                  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

■

**Satz 4.7.** Der Greedy Algorithmus garantiert für jede Folge von Verlusten aus  $\{0, 1\}$

$$L_G^T \leq N \cdot L_{min}^T + (N - 1).$$

*Beweis.* Teile die Zeit in Phasen  $0, \dots, L_{min}^T$  ein, so dass gilt:

Jeder Schritt  $t$  mit den minimalen Kosten eines Experten  $L_{min}^{t-1} = i$  gehört zu Phase  $i$ . In jeder Phase  $i < L_{min}^T$  erfährt der Greedy-Algorithmus einen Verlust von höchstens  $N$ , wenn zu Beginn der Phase jeder Experte minimale Kosten hatte und in jedem Schritt nur ein Experte seine Kosten um 1 erhöht. In Phase  $L_{min}^T$  ist der Verlust von Greedy höchstens  $N - 1$ . □

**Satz 4.8.** Für jeden deterministischen Online-Algorithmus  $D$  und jedes  $T \geq 1$  gibt es eine Folge von  $T$  Verlusten, so dass

$$L_D^T = T \quad \text{und} \quad L_{min}^T \leq \lfloor T/N \rfloor.$$

Diese untere Schranke kann man leicht zeigen, indem man das obige Beispiel für Greedy verallgemeinert. Durch den Determinismus weiß der Gegner immer, welchen Experten der Algorithmus als nächstes auswählen wird. Genau diesem Experten werden in der kommenden Runde Kosten 1 zugewiesen, den anderen Kosten 0.

Die untere Schranke zeigt, dass man ohne Randomisierung die Garantie des Greedy Algorithmus nicht schlagen kann.

Daher betrachten wir eine randomisierte Version des Greedy-Algorithmus.

---

**Algorithmus 3:** Randomisierter Greedy-Algorithmus (RG)

---

**for** jeden Schritt  $t$  **do**

Sei  $S^{t-1} = \{i : L_i^{t-1} = L_{min}^{t-1}\}$   
 Setze  $p_i^t = 1/|S^{t-1}|$  für jedes  $i \in S^{t-1}$ , und sonst  $p_i^t = 0$ .

---

In Worten: In jedem Schritt  $t$  wählt RG einen Experten aus der Menge  $S^{t-1}$  unabhängig und uniform zufällig.

**Satz 4.9.** Der RG Algorithmus garantiert für jede Folge von Verlusten aus  $\{0, 1\}$

$$L_{RG}^T \leq (1 + \ln N)L_{min}^T + \ln N.$$

*Beweis.* Betrachte Phase  $i < L_{min}^T$  (Phase definiert wie oben). Sei  $u$  der erste Schritt und  $v$  der letzte Schritt dieser Phase. In jedem Schritt  $t \in \{u, \dots, v\}$  wählt RG einen Experten aus  $S^{t-1}$  unabhängig und uniform zufällig. Dabei gilt  $[N] \supseteq S^{u-1} \supseteq S^u \supseteq \dots \supseteq S^{v-1} \supseteq \emptyset$ .

Für eine einfachere Notation setzen wir  $S^v = \emptyset$ . Der Verlust von RG in der betrachteten Phase beträgt

$$\sum_{t=u}^v \frac{|S^{t-1}| - |S^t|}{|S^{t-1}|},$$

da  $S^{t-1} \setminus S^t$  die Experten aus  $S^{t-1}$  mit Verlust 1 in Schritt  $t$  sind, und die Experten in  $S^t$  einen Verlust 0 erfahren. Dieser Ausdruck wird maximiert, wenn  $|S^{t-1} \setminus S^t| = 1$  für  $u \leq t \leq v$ . Dann ist die Summe

$$\sum_{t=1}^{|S^{u-1}|} \frac{1}{t} \leq 1 + \ln |S^{u-1}| \leq 1 + \ln N.$$

Der Verlust von RG in Phase  $i < L_{min}^T$  ist höchstens  $1 + \ln N$ . Genauso ist der Verlust von RG in Phase  $L_{min}^T$  höchstens  $\sum_{t=2}^N \frac{1}{t} \leq \ln N$ .

Daraus folgt der Satz. □

Dieses Ergebnis kann noch weiter verbessert werden mit dem Randomized Weighted Majority Algorithmus.

**Algorithmus 4:** Randomized Weighted Majority Algorithmus (RMW)

---

Sei  $\eta \in (0, \frac{1}{2}]$  ein passend gewählter Parameter.  
 Am Anfang sei  $w_i^1 = 1$  für jedes  $i \in [N]$ .  
**for** jeden Schritt  $t$  **do**  
   Sei  $W^t = \sum_{i=1}^N w_i^t$   
   Wähle Experten  $i$  mit Wahrscheinlichkeit  $p_i^t = w_i^t/W^t$   
   Setze  $w_i^{t+1} = w_i^t \cdot (1 - \eta)^{\ell_i^t}$

---

RWM besitzt eine Wahrscheinlichkeitsverteilung über die gegebenen Experten und passt diese über die Zeitschritte an. In jedem Schritt werden die Gewichte  $w_i^{t+1}$  in Abhängigkeit der Kosten von  $i$  in Runde  $t$  verringert, so dass Experten mit niedrigen Kosten langfristig mit höherer Wahrscheinlichkeit gewählt werden. Trotzdem gilt immer  $w_i^{t+1} > 0$ , damit die Randomisierung auch nach langer Laufzeit gewahrt bleibt. Der Parameter  $\eta$  beeinflusst wie schnell der Algorithmus auf die bisherigen Kosten reagiert. Wenn der Parameter sehr klein gewählt ist, dauert der Lernprozess sehr lange. Ist der Parameter sehr groß, reagiert der Algorithmus schnell, aber kann dadurch eher vom Gegner vorhergesagt werden.

**Satz 4.10** (Littlestone, Warmuth, 1994). *Der RWM Algorithmus garantiert für jede Folge von Verlusten aus  $[0, 1]$*

$$L_{RWM}^T \leq (1 + \eta)L_{min}^T + \frac{\ln N}{\eta} .$$

Wenn wir  $\eta = \sqrt{\frac{\ln N}{T}}$  wählen, dann ergibt sich

$$L_{RWM}^T \leq L_{min}^T + 2\sqrt{T \ln N} .$$

Der Regret eines Lernalgorithmus  $H$  wird definiert als  $R(T) = L_H^T - L_{min}^T$ .

**Korollar 4.11.** *Der RWM Algorithmus mit  $\eta = \sqrt{\frac{\ln N}{T}}$  hat einen Regret von höchstens  $2\sqrt{T \ln N}$ .*

Der durchschnittliche Regret pro Zeitschritt ist daher nur  $2\sqrt{\frac{\ln N}{T}}$ .

Beachte: Dieser Term geht gegen 0 für wachsendes  $T$ .

**Definition 4.12** (No-Regret Algorithmus). Sei  $H$  ein Online-Lernalgorithmus. Der Regret von  $H$  bis zum Zeitpunkt  $T$  beträgt

$$R(T) = L_H^T - L_{min}^T .$$

Bei einem No-Regret Algorithmus gilt für den durchschnittlichen Regret pro Zeitschritt

$$\frac{R(T)}{T} \longrightarrow 0 \quad \text{für } T \rightarrow \infty .$$

Algorithmen mit dieser Eigenschaft heißen No-Regret Lernalgorithmen.

Im Gegensatz zum einfachen Greedy ist RWM ein No-Regret Algorithmus.

*Beweis (Satz 4.10).* Wir analysieren, wie sich die Summe der Gewichte  $W^t$  über die Zeit verringert. Es gilt

$$W^{t+1} = \sum_{i=1}^N w_i^{t+1} = \sum_{i=1}^N w_i^t (1 - \eta)^{\ell_i^t} .$$

Wir sehen, dass  $(1 - \eta)^\ell = (1 - \ell\eta)$  in beiden Fällen  $\ell = 0$  und  $\ell = 1$  gilt. Für den Bereich  $\ell \in [0, 1]$  Daneben ist  $(1 - \eta)^\ell$  eine konvexe Funktion in  $\ell$ . Zusammen genommen resultiert daraus, dass  $(1 - \eta)^\ell \leq (1 - \ell\eta)$ . Aus dieser Einsicht folgt:

$$W^{t+1} \leq \sum_{i=1}^N w_i^t (1 - \ell_i^t \eta) .$$

Sei  $F^t$  der erwartete Verlust von RWM in Zeitschritt  $t$ , dann gilt  $F^t = \sum_{i=1}^N \ell_i^t w_i^t / W^t$ . Das Einsetzen in die Schranke für  $W^{t+1}$  ergibt

$$W^{t+1} \leq W^t - \eta F^t W^t = W^t (1 - \eta F^t) .$$

Daraus folgt

$$W^{T+1} \leq W^1 \prod_{t=1}^T (1 - \eta F^t) = N \prod_{t=1}^T (1 - \eta F^t) .$$

Die Summe der Gewichte nach Schritt  $T$  kann also nach oben beschränkt werden durch den erwarteten Verlust von RWM.

Andererseits kann die Summe der Gewichte nach Schritt  $T$  auch nach unten beschränkt werden durch den Verlust des besten Experten:

$$W^{T+1} \geq \max_{1 \leq i \leq N} (w_i^{T+1}) = \max_{1 \leq i \leq N} \left( (1 - \eta)^{\sum_{t=1}^T \ell_i^t} \right) = (1 - \eta)^{L_{min}^T} .$$

Die Kombination der beiden Schranken und Logarithmieren auf beiden Seiten führt zu

$$L_{min}^T \ln(1 - \eta) \leq (\ln N) + \sum_{t=1}^T \ln(1 - \eta F^t) .$$

Wir vereinfachen den Ausdruck mit Hilfe der folgenden Abschätzung

$$-z - z^2 \leq \ln(1 - z) \leq -z ,$$

die für jedes  $z \in [0, \frac{1}{2}]$  gilt. Diese Vereinfachung ergibt

$$L_{min}^T (-\eta - \eta^2) \leq (\ln N) + \sum_{t=1}^T (-\eta F^t) = (\ln N) - \eta L_{RWM}^T .$$

Zum Schluß lösen wir die Ungleichung nun nur noch nach  $L_{RWM}^T$  auf:

$$L_{RWM}^T \leq (1 + \eta) L_{min}^T + \frac{\ln N}{\eta} .$$

□

Regret-Minimierung ist ein natürlicher Ansatz für Verhalten bei wiederholten Entscheidungen mit unvollständiger Information.

Wir betrachten Regret-Lernen in einem Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (c_i)_{i \in \mathcal{N}})$ , das  $T$  Runden lang wiederholt hintereinander gespielt wird. (sog. *wiederholtes Spiel*).

Anfangs kennt keiner der Spieler  $i \in \mathcal{N}$  das Spiel. In jeder Runde  $t$  wählt jeder Spieler  $i$  eine reine Strategie  $s_i^t \in \Sigma_i$  mit seinem eigenen No-Regret Algorithmus. Der Algorithmus von  $i$  benutzt nur die *beobachteten Kosten von  $i$  in den vorherigen Runden*.

Betrachte einen Ablauf des Spiels  $s^1, s^2, \dots, s^T$  über  $T$  Runden. Wir interpretieren ihn als Verteilung über Zustände, in dem wir  $k \in [T]$  uniform zufällig wählen.

Wenn Spieler  $i$  einen Regret von  $R_i(T)$  hat, dann gilt für jede Strategie  $s'_i \in S_i$

$$\begin{aligned} \mathbb{E}_{k \in [T]} [c_i(s^k)] &= \sum_{t=1}^T \frac{1}{T} \cdot c_i(s^t) \leq \sum_{t=1}^T \frac{1}{T} \cdot c_i(s'_i, s_{-i}^t) + \frac{R_i(T)}{T} \\ &= \mathbb{E}_{k \in [T]} [c_i(s'_i, s_{-i}^k)] + \frac{R_i(T)}{T} . \end{aligned}$$

**Proposition 4.13.** *Wenn jeder Spieler nach  $T$  Runden Regret höchstens  $R$  hat, dann stellt der Ablauf des Spiels ein  $\frac{R}{T}$ -approximatives grob-korreliertes Gleichgewicht dar.*

Wenn alle Spieler RWM nutzen, dann wird schon nach  $T = \frac{4}{\varepsilon^2} \cdot \log(\max_i |S_i|)$  Runden ein  $\varepsilon$ -approximatives grob-korreliertes Gleichgewicht erreicht.

### 4.1.3 Nullsummenspiele

In der obigen Analyse haben wir ein Gegnermodell verwendet, bei dem der Gegner die Kosten der Algorithmen möglichst verschlechtert. Expliziter betrachten einen Spieler, der über die Zeitschritte lernt, und einen Gegner, der die Kosten möglichst erhöht. Dieses Konzept kennen wir aus den Nullsummenspielen, hier erhöhen die Spieler ihren Nutzen, je mehr sie ihrem Gegner schaden.

Im Gegnermodell kennen beide Spieler das Spiel nicht, in dem sie spielen. Sie verwenden No-Regret Algorithmen für die Wahl der Strategie.

Können die Spieler optimale Strategien und somit ein gemischtes Nash-Gleichgewicht *erlernen*?

Wir betrachten Spieler II, der Experten (= reine Strategien) wählen muss, wobei Spieler I sein Gegner ist. In jedem Schritt  $t$  wählt der Lernalgorithmus  $H$  von Spieler II eine gemischte Strategie  $y^t$  gegen die unbekannte Strategie  $x^t$  des Gegenspielers I. Der Verlust in Schritt  $t$  für die Strategie (Experte)  $i$  beträgt

$$\ell_i^t = \sum_{j \in \Sigma_I} x_j^t a_{ji} .$$

Der gesamte Verlust des Lernalgorithmus  $H$  in Schritt  $t$  beträgt

$$\ell_H^t = c_{II}(x^t, y^t) = \sum_{i \in \Sigma_{II}} \sum_{j \in \Sigma_I} x_j^t a_{ji} y_i^t .$$

Sei  $H$  ein No-Regret Lernalgorithmus, so dass

$$\frac{L_H^T - L_{min}^T}{T} \rightarrow 0 \quad \text{für } T \rightarrow \infty .$$

Der durchschnittliche Verlust pro Schritt eines No-Regret Algorithmus wird so klein wie der durchschnittliche Verlust der (im Nachhinein nach  $t$  Schritten) *besten reinen Strategie*.

Wird der durchschnittliche Verlust  $L_H^T/T$  so klein wie der Wert des Spiels?

**Satz 4.14.** *Betrachte ein wiederholtes Nullsummenspiel mit 2 Spielern. Wenn Spieler II über die  $T$  Zeitschritte einen Algorithmus  $H$  mit Regret  $R$  für seine Strategiewahl nutzt, dann beträgt sein durchschnittlicher Verlust*

$$\frac{L_H^T}{T} \leq v_I^* + \frac{R}{T} .$$

*Dies Resultat gilt analog für Spieler I und die Loss-Ceiling.*

*Beweis.* Wir zeigen, dass die beste Strategie am Ende der  $T$  Schritte die Gesamtkosten von höchstens  $L_{min}^T \leq T \cdot v_I^*$  hat.

Betrachte den Ablauf der Strategiewahlen des Gegenspielers I also Strategien  $x^1, x^2, \dots, x^T$ , und addiere sie zu einer "durchschnittlichen Strategie"

$$\hat{x}_j = \frac{1}{T} \sum_{t=1}^T x_j^t \quad \text{für jedes } j \in \Sigma_I .$$

Wegen der Linearität der Kosten bleibt der gesamte Verlust  $L_i^T$  einer reinen Strategie  $i \in \Sigma_{II}$  gleich, wenn Spieler I in allen Zeitschritten immer nur  $\hat{x}$  gespielt hätte:

$$L_i^T = \sum_{t=1}^T \sum_{j \in \Sigma_I} x_j^t \cdot a_{ji} = \sum_{j \in \Sigma_I} \left( \sum_{t=1}^T x_j^t \right) \cdot a_{ji} = T \cdot \sum_{j \in \Sigma_I} \hat{x}_j \cdot a_{ji} .$$

Nehmen wir also an, I spielt immer nur  $\hat{x}$  und betrachten die im Nachhinein beste reine Strategie für II. Damit ergibt sich nun ein Spiel mit nur noch einem Zeitschritt.

In diesem Einschnittspiel muss Spieler I zuerst die durchschnittliche Strategie  $\hat{x}$  wählen, dann

wählt Spieler II eine (reine) beste Antwort  $\hat{x}$  – also muss I zuerst wählen, dann antwortet II. Per Definition des Gain-Floors gibt es immer  $i \in \Sigma_{\text{II}}$ , so dass der Nutzen von I (und der Verlust von II) höchstens  $v_{\text{I}}^*$  beträgt, d.h.,  $c_{\text{II}}(\hat{x}, i) \leq v_{\text{I}}^*$ . Also gibt es eine reine Strategie  $i \in \Sigma_{\text{II}}$  mit

$$L_{\min}^T \leq L_i^T \leq T \cdot v_{\text{I}}^* .$$

Zusammen ergeben die beiden Einsichten:

$$L_H^T \leq L_{\min}^T + R \leq T \cdot v_{\text{I}}^* + R .$$

□

Diese Erkenntnis wenden wir für einen alternativen Beweis des Minimax Theorems an.

**Satz 3.28** (Minimax Theorem). *In jedem Nullsummenspiel mit 2 Spielern gilt  $v = v_{\text{I}}^* = v_{\text{II}}^*$ .*

*Beweis.* Zum Widerspruch nehmen wir an, dass  $v_{\text{I}}^* + \gamma = v_{\text{II}}^*$  für ein  $\gamma > 0$ . Beide Spieler spielen nun das Spiel wiederholt mit einem Lernalgorithmus und sei  $T$  groß genug, so dass der Regret der Algorithmen  $R/T < \gamma/3$ . Über die Gesamtzeit der  $T$  Runden können wir wie oben  $L_{\min}^T \leq v_{\text{I}}^*$  für Spieler II, und  $L_{\min}^T \leq -v_{\text{II}}^*$  für Spieler I (“–” da Verlust) zeigen. Dies bedeutet, dass die Algorithmen höchstens  $v_{\text{I}} + \gamma/3$  durchschnittliche Kosten für Spieler II und mindestens  $v_{\text{II}} - \gamma/3$  durchschnittlichen Nutzen für Spieler I garantieren. Allerdings entsprechen die durchschnittlichen Kosten für II dem durchschnittlichen Nutzen für I

→ Widerspruch. □

**Korollar 4.15.** *Wenn beide Spieler einen No-Regret Lernalgorithmus nutzen, konvergiert der durchschnittliche Ablauf des Spiels  $(\hat{x}, \hat{y})$  zu optimalen Strategien und damit zu einem gemischten Nash-Gleichgewicht des Spiels.*

Der Satz garantiert Konvergenz nur für den *durchschnittlichen Ablauf des Spiels*, aber nicht für das *eigentliche Verhalten* in den Strategien  $x^t$  und  $y^t$ !

**Satz 4.16.** *Es gibt No-Regret Lernalgorithmen für Spieler I und II mit denen die Strategien  $x^t$  und  $y^t$  nicht zu optimalen Strategien konvergieren.*

**Beispiel 4.17.** *Matching Pennies (normalisiert)*

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

■

*Beweis.* Wir betrachten einen “seltsamen” No-Regret Algorithmus  $H$  für Matching Pennies: I und II wählen in jedem Schritt reine Strategien, dabei wechselt I zur anderen reinen Strategie in Runde 1, 3, 5, ..., II wechselt zur anderen reinen Strategie in Runde 2, 4, 6, ... Wenn ein Spieler von der Regel abweicht, ruft der andere Spieler den RWM-Algorithmus auf. (Dieser Trick sichert die No-Regret-Eigenschaft für *jeden möglichen* Ablauf).

Der durchschnittliche Verlust  $L_i^T/T \rightarrow 0.5$  für beide Strategien  $i = 1, 2$  von II entspricht dem durchschnittlichen Verlust des Algorithmus  $L_H^T/T \rightarrow 0.5$ .  $H$  ist also ein *No-Regret Algorithmus für II!* (gleiches Argument für I) Trotzdem ist keine der Verteilungen  $y^t$  nahe an der optimalen Strategie (0.5, 0.5) und kein Verlust in einer einzelnen Runde  $\ell_H^t$  ist nahe am Wert  $v = 0.5$ . □

Durch die kreisenden Abweichungen des Algorithmus  $H$  werden die optimalen Strategien nur im Durchschnitt über die Zeit erreicht. Es gibt eine Variante des RWM Algorithmus, der ein solches Verhalten ausschließt.

---

**Algorithmus 5:** Variabler Randomized Weighted Majority (vRWM) Algorithmus (Freund, Schapire, 1999)

---

Sei  $\eta_0 \in (0, \frac{1}{2}]$  und  $u$  eine obere Schranke auf den Wert des Spiels.

Initialisiere  $w_i^1 = 1$  für alle  $i \in [N]$

**for** jeden Schritt  $t$  **do**

Sei  $W^t = \sum_{i=1}^N w_i^t$

Wähle Experte  $i$  mit W.keit  $p_i^t = w_i^t/W^t$

**if**  $\ell_{vRWM}^t \leq u$  **then**

Setze  $w_i^{t+1} = w_i^t$

**else**

Setze  $\eta_t = 1 - \frac{u(1-\ell_{vRWM}^t)}{(1-u)\ell_{vRWM}^t}$

Und  $w_i^{t+1} = w_i^t \cdot (1 - \eta_t)^{\ell_i^t}$

**Definition 4.18** (Relative Entropie). Die *Kullback-Leibler Divergenz* oder *relative Entropie* von zwei Verteilungen  $y$  und  $y'$  ist gegeben durch

$$RE(y \parallel y') = \sum_{i=1}^n y_i \cdot \ln \left( \frac{y_i}{y'_i} \right) .$$

Zum Vergleich von  $a, b \in [0, 1]$  nutzen wir die Verteilungen  $(a, 1 - a)$  und  $(b, 1 - b)$ :

$$\begin{aligned} RE(a \parallel b) &= RE((a, 1 - a) \parallel (b, 1 - b)) \\ &= a \ln \left( \frac{a}{b} \right) + (1 - a) \ln \left( \frac{1 - a}{1 - b} \right) . \end{aligned}$$

Die Kullback-Leibler Divergenz für Verteilungen ist *immer nicht-negativ* und  $RE(y \parallel y') = 0$  genau dann, wenn  $y = y'$ .

In jedem Schritt, in dem der Verlust von vRWM zu hoch ist, bewegt die Anpassung die neue Strategie näher an jede gute Strategie. Durch die relative Entropie kann der aktuelle Grad der Anpassung ausgedrückt werden.

**Satz 4.19.** Sei  $y'$  eine beliebige gemischte Strategie für  $\Pi$ , die einen Verlust von höchstens  $u$  garantiert gegen jede beste Antwort von  $\mathbf{I}$ . In jeder Iteration  $t$  von vRWM, in der  $\ell_{vRWM}^t \geq u$ , sinkt die relative Entropie zwischen  $y'$  und  $y^{t+1}$  um

$$RE(y' \parallel y^{t+1}) \leq RE(y' \parallel y^t) - RE(u \parallel \ell_{vRWM}^t) .$$

*Beweis.* Zur Vollständigkeit beweisen wir den Satz. Betrachte einen Schritt  $t$  mit  $\ell_{vRWM}^t > u$ . Dann gilt

$$\begin{aligned} RE(y' \parallel y^{t+1}) - RE(y' \parallel y^t) &= \sum_{i \in \Sigma_{\Pi}} y'_i \ln \frac{y'_i}{y_i^{t+1}} - \sum_{i \in \Sigma_{\Pi}} y'_i \ln \frac{y'_i}{y_i^t} \\ &= \sum_{i \in \Sigma_{\Pi}} y'_i \ln \frac{y_i^t}{y_i^{t+1}} \\ &\leq \sum_{i \in \Sigma_{\Pi}} y'_i \ln \frac{1 - \eta_t \ell_{vRWM}^t}{(1 - \eta_t)^{\ell_i^t}} . \end{aligned}$$

Wir verwenden, dass  $(1 - \eta)^\ell = (1 - \ell\eta)$  in beiden Fällen  $\ell = 0$  und  $\ell = 1$ . Daneben ist  $(1 - \eta)^\ell$  eine konvexe Funktion in  $\ell$ , und für  $\ell \in [0, 1]$  ergibt dies  $(1 - \eta)^\ell \leq (1 - \ell\eta)$ . Daraus folgt, dass  $W^{t+1} = \sum_{i=1}^N w_i^{t+1} = \sum_{i=1}^N w_i^t (1 - \eta)^{\ell_i^t} \leq \sum_{i=1}^N w_i^t (1 - \ell_i^t \eta)$ . Sei  $F^t$  der erwartete Verlust von RWM in Zeitschritt  $t$ , dann gilt  $F^t = \sum_{i=1}^N \ell_i^t w_i^t / W^t$ . Das Einsetzen in die Schranke für  $W^{t+1}$  ergibt

$$W^{t+1} \leq W^t - \eta F^t W^t = W^t (1 - \eta F^t) .$$

Es gilt

$$W^{T+1} \leq W^1 \prod_{t=1}^T (1 - \eta F^t) = N \prod_{t=1}^T (1 - \eta F^t) .$$

Zusammengefasst wissen wir, dass  $y_i^{t+1} = w_i^t (1 - \eta_t)^{\ell_i^t} / W^{t+1}$  und  $W^{t+1} \leq W^t (1 - \eta_t F^t) = W^t (1 - \eta_t \ell_{vRW}^t)$ .

$$\begin{aligned} RE(y' \parallel y^{t+1}) - RE(y' \parallel y^t) &\leq \sum_{i \in \Sigma_{\text{II}}} y'_i \ln \frac{1 - \eta_t \ell_{vRW}^t}{(1 - \eta_t)^{\ell_i^t}} \\ &= \sum_{i \in \Sigma_{\text{II}}} y'_i \ln \left( \frac{1}{1 - \eta_t} \right)^{\ell_i^t} + \ln(1 - \eta_t \ell_{vRW}^t) \\ &= \left( \ln \frac{1}{1 - \eta_t} \right) \cdot \sum_{i \in \Sigma_{\text{II}}} y'_i \ell_i^t + \ln(1 - \eta_t \ell_{vRW}^t) \\ &\leq \left( \ln \frac{1}{1 - \eta_t} \right) u + \ln(1 - \eta_t \ell_{vRW}^t) , \end{aligned}$$

weil die Strategie  $y'$  nie mehr als  $u$  an Verlust erzeugt. Die Ableitung von

$$\left( \ln \frac{1}{1 - \eta_t} \right) u + \ln(1 - \eta_t \ell_{vRW}^t)$$

nach  $\eta_t$  setzen wir gleich 0. Damit erhalten wir das Minimum

$$\eta_t = 1 - \frac{u(1 - \ell_{vRW}^t)}{(1 - u)\ell_{vRW}^t}$$

wie gewünscht. Einsetzen in der Formel ergibt

$$-u \ln \left( \frac{u}{\ell_{vRW}^t} \cdot \frac{1 - \ell_{vRW}^t}{1 - u} \right) + \ln \frac{1 - \ell_{vRW}^t}{1 - u} = -RE(u \parallel \ell_{vRW}^t) .$$

□

**Korollar 4.20.** *Für jede Folge von Strategien  $x^1, x^2, \dots$  beträgt die Anzahl der Runden mit Verlust  $\ell_{vRW}^t \geq u + \varepsilon$  höchstens*

$$\frac{\ln |\Sigma_{\text{II}}|}{RE(u \parallel u + \varepsilon)} .$$

Für ein festes  $\varepsilon$  ist diese Zeit unabhängig von  $T$ . Für spätere Zeitschritte  $t$  muss daher der Verlust immer näher an  $u$  herankommen. Dieses Verhalten ist viel besser als z.B. beim seltsamen No-Regret Algorithmus, der einen Verlust von 1 in jeder zweiten Runde generiert, sogar für beliebig späte Zeitschritte  $t$ .

### 4.1.4 Konkave Spiele

Zusätzlich zu Nullsummenspielen gibt es weitere interessante Klassen von Spielen wie konkave Spiele, bei denen gemischte Nash-Gleichgewichte mit No-Regret Algorithmen erlernt werden können.

#### Beispiel 4.21. Bandbreitenspiel

Eine Menge  $\mathcal{N}$  von  $n$  Nutzern möchten Videos über einen gemeinsamen Internetzugang mit Kapazität  $C$  (sei o.B.d.A.  $C = 1$ ) herunterladen. Als Strategie platziert Spieler  $i$  ein Gebot  $s_i \in \Sigma_i$ , wobei  $\Sigma_i = [b_{\min}, 1]$  mit  $b_{\min} > 0$  als Minimalgebot.

Der Service-Provider  $M$  sammelt den Vektor  $s$  aller Gebote und richtet eine proportionale Durchsatzrate für jeden Spieler ein:

$$M_i(s) = \frac{s_i}{\sum_{j \in \mathcal{N}} s_j}.$$

Beachte, dass  $M_i(s) > 0$  und  $\sum_i M_i(s) = 1 = C$ .

Spieler  $i$  erhält seine Rate und zahlt sein Gebot an den Provider mit dem Nutzen:

$$u_i(s) = \alpha_i \cdot M_i(s) - s_i = \frac{\alpha_i \cdot s_i}{\sum_{j \in \mathcal{N}} s_j} - s_i.$$

(Spieler  $i$  tauscht Geld gegen Rate mit Faktor  $\alpha_i > 0$ ).

Sei in diesem Beispiel  $b_{\min} = 0.01$ .

| Spieler | $\alpha_i$ | Gebot $s_i$ | Rate $M_i(s)$ | Nutzen $u_i(s)$     |
|---------|------------|-------------|---------------|---------------------|
| 1       | 2          | 0.9         | 0.45          | $0.90 - 0.9 = 0$    |
| 2       | 3          | 0.7         | 0.35          | $1.05 - 0.7 = 0.35$ |
| 3       | 4          | 0.4         | 0.25          | $1.00 - 0.4 = 0.60$ |

Die konkaven Nutzenfunktionen  $u_i(s_i, s_{-i})$  für  $s = (0.9, 0.7, 0.4)$  sind hier:

- $u_1(x, 0.7, 0.4) = \frac{2x}{x+1.1} - x$
- $u_2(0.9, x, 0.4) = \frac{3x}{x+1.3} - x$
- $u_3(0.9, 0.7, x) = \frac{4x}{x+1.6} - x$

■

**Definition 4.22** (Konkav/Konvex). Eine Funktion  $f : X \rightarrow \mathbb{R}$  ist *konvex* (*konkav*) auf  $X \subset \mathbb{R}^k$ , wenn die direkte Verbindung zwischen  $f(x)$  und  $f(y)$  immer *über* (*unter*)  $f$  liegt,  $\forall x, y \in X$ .

Formal sei für alle  $x, y \in \mathbb{R}$

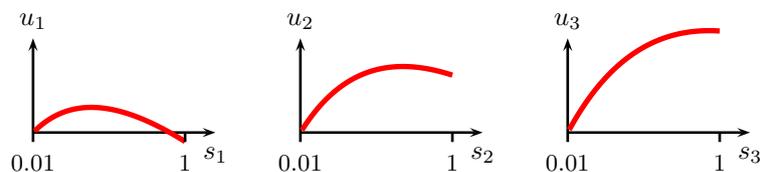
$$\text{Konvex: } \lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y), \quad \text{für jedes } \lambda \in (0, 1)$$

$$\text{Konkave: } \lambda f(x) + (1 - \lambda)f(y) \leq f(\lambda x + (1 - \lambda)y), \quad \text{für jedes } \lambda \in (0, 1).$$

Für die Definition von streng konkav/konvex müssen  $\geq$  und  $\leq$  in der Definition ersetzt werden durch  $>$  und  $<$ .

Wenn  $f$  (streng) konkav ist, dann ist  $-f$  (streng) konvex und umgekehrt.

Die Nutzenfunktionen im Bandbreitenspiel sind streng konkav. Im obigen Beispiel:



Ein Bandbreitenspiel ist kein endliches Spiel, es gibt unendlich viele Strategien. Gibt es immer ein Nash-Gleichgewicht?

Mit der Konkavität der Nutzenfunktionen und dem Satz von Brouwer kann man zeigen:

**Lemma 4.23.** *Jedes Bandbreitenspiel hat mindestens ein (reines) Nash-Gleichgewicht.*

Wie kann sich in diesen Spielen ein Nash-Gleichgewicht einstellen?  
Konvergieren die Spieler mit No-Regret Lernalgorithmen zum Nash-Gleichgewicht?

Dafür benötigen wir zuerst No-Regret-Algorithmen für unendlich viele Experten.

Wir definieren ein *Experten-Problem mit unendlich vielen Experten*:

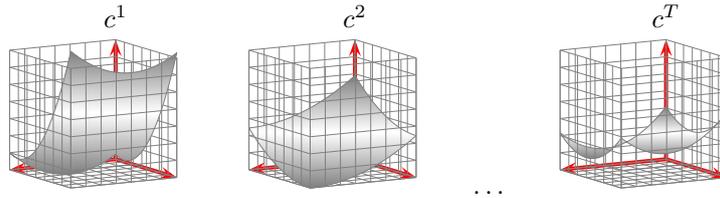
Alle Punkte einer *konvexen und kompakten Menge*  $D \subset \mathbb{R}^k$  sind "Experten". In jedem Schritt  $t = 1, \dots, T$  wählen wir einen Punkt  $x^t \in D$ . Danach wird eine *differenzierbare und konvexe* Kostenfunktion  $c^t : D \rightarrow \mathbb{R}$  aufgedeckt.

Das Ziel ist es,  $x^t$ 's so zu wählen, dass die Gesamtkosten  $\sum_{t=1}^T c^t(x^t)$  minimiert werden.

Beachte, dass anstatt der Minimierung von konvexen Kosten genauso die Maximierung von konkaven Nutzenfunktionen  $-c^t(x^t)$  betrachten könnten.

**Beispiel 4.24.** *Wiederholtes Bandbreitenspiel*

Hier ist z.B.  $D = [0, 1]^2$ , und wir sehen diverse konvexe Kostenfunktionen:



Jeder Spieler  $i$  hat eine kompakte und konvexe Strategiemenge  $D = [b_{\min}, 1]$ . Die Nutzenfunktion  $u_i(s_i^t, s_{-i}^t)$  ist konkav und differenzierbar. Dazu hängt sie von unbekanntem Geboten  $s_{-i}^t$  ab, die nur bekannt werden, nachdem Strategie  $s_i^t$  gewählt wurde. ■

In konvexen Spielen definieren wir

- die Kosten des Entscheiders:  $\sum_{t=1}^T c^t(x^t)$ ,
- wobei der beste Experte  $x^* \in \arg \min_{x \in D} \sum_{t=1}^T c^t(x)$ .
- Der durchschnittliche Regret pro Zeitschritt beträgt

$$\frac{R(T)}{T} = \frac{1}{T} \left( \sum_{t=1}^T c^t(x^t) - \sum_{t=1}^T c^t(x^*) \right) .$$

Ein Algorithmus besitzt die *No-Regret Eigenschaft*, wenn  $\frac{R(T)}{T} \rightarrow 0$  für  $T \rightarrow \infty$ .

Wir entwerfen einen No-Regret-Algorithmus mit projiziertem Gradientenabstieg.

**Definition 4.25** (Gradient). Der *Gradient* einer differenzierbaren Funktion  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  am Punkt  $x \in \mathbb{R}^k$  ist ein Vektor  $\nabla f(x) \in \mathbb{R}^k$ , der in die Gegenrichtung des stärksten Abstiegs zeigt.

Im Bandbreitenspiel sind die Kosten  $c^t(x) = -u_i(x, s_{-i}^t)$  nur eine Funktion von  $x \in \mathbb{R}$ , da  $s_{-i}^t$  nicht vom Spieler  $i$  beeinflusst werden kann. Dann ist  $\nabla c^t$  die Ableitung von  $c^t$  nach  $x$ :

$$\nabla c^t(x) = 1 - \alpha_i \cdot \frac{\sum_{j \neq i} s_j^t}{\left(x + \sum_{j \neq i} s_j^t\right)^2} .$$

Um eine *einzig*e konvexe Funktion  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  zu minimieren, reicht es aus von einem beliebigen Startpunkt  $x^1 \in \mathbb{R}^k$  mit Hilfe des Gradienten nacheinander kleine Schritte in die Richtung des stärksten Abstiegs zu machen

$$x^{t+1} = x^t - \eta \cdot \nabla f(x^t) .$$

Mit hinreichend kleiner Schrittlänge  $\eta \in \mathbb{R}$  "sinken wir in das Tal" der konvexen Funktion und erreichen die Nähe (je nach  $\eta$ ) des globalen Minimums.

Der Gradientenabstieg mit einer einzigen konvexen Funktion  $f$  über einem konvexen Raum  $D \subset \mathbb{R}^k$  kann uns aus  $D$  herausführen. Wir nutzen einen Trick, um das zu verhindern: Wir schieben das Ergebnis zurück nach  $D$  wie folgt.

**Definition 4.26** (Projektion). Wir definieren eine *Projektion*  $P : \mathbb{R}^k \rightarrow D$  mit  $P(x) = x'$ , wobei  $x \in \mathbb{R}^k$  beliebig und  $x' \in D$  der Punkt aus  $D$  mit kürzester Distanz zu  $x$ .

Im Bandbreitenspiel gilt  $P(x) = x$  für  $x \in [b_{\min}, 1]$ ,  $P(x) = 1$  für  $x > 1$ , und  $P(x) = b_{\min}$  für  $x < b_{\min}$ .

Die Projektion wirft einen Punkt  $x$  zurück auf den nächstgelegenen Punkt in  $D$ . Wir minimieren eine einzige konvexe Funktion  $f$  über einen konvexen und kompakten Raum  $D$  mit projiziertem Gradientenabstieg

$$x^{t+1} = P(x^t - \eta \nabla f(x^t)) .$$

Durch die Konvexität "laufen wir am Rand von  $D$  entlang" und sinken so zum Minimum von  $f$  in  $D$ .

Im Online-Fall mit wechselnden Funktionen über  $T$  Runden nutzen wir den *Generalized Infinitesimal Gradient Ascent* oder GIGA Algorithmus:

$$\text{Wähle } x^1 \in D \text{ beliebig} \quad \text{und} \quad x^{t+1} = P(x^t - \eta \cdot \nabla c^t(x^t)) .$$

GIGA nutzt den Gradienten von  $c^t$  zur Optimierung von  $c^{t+1}$ . Das wirkt wie eine ziemlich schlechte Idee, denn  $c^{t+1}$  kann sich stark von  $c^t$  unterscheiden. Dennoch gilt:

**Satz 4.27** (Zinkevich, 2003). Sei  $G \geq \|\nabla c^t(x)\|$  und  $\Delta \geq \|x - y\|$  für jedes  $x, y \in D$  und  $t = 1, \dots, T$ . Wenn  $\eta = \frac{\Delta}{G\sqrt{T}}$ , dann erzeugt GIGA einen Regret von

$$R(T) \leq \Delta \cdot G \cdot \sqrt{T} .$$

Wenn  $G$  und  $\Delta$  unabhängig von  $T$  sind, dann ist GIGA ein No-Regret Algorithmus.

*Beweis.* Intuitiv "funktioniert" GIGA gut, wenn alle Kostenfunktionen  $c^t$  des Projizierten Gradientenabstiegs ähnlich sind. Wenn die Funktionen sehr unterschiedlich sind, erzeugt GIGA hohe Kosten, aber dann muss auch das Optimum  $x^* \in D$  hohe Kosten haben.

Wir erfassen diese Intuition mit einem Potenzialargument:

OBdA. sei das Optimum  $x^* = 0$ , der Ursprung eines Koordinatensystems. Betrachte das "Potential" als  $\Phi_t = \frac{1}{2\eta} \|x^t\|^2$ , wobei  $\Phi_t$  die Distanz von  $x^t$  zu  $x^* = 0$  liefert.

**Lemma 4.28** (Kosten-gegen-Distanz).

$$c^t(x^t) - c^t(0) + \Phi_{t+1} - \Phi_t \leq \eta \cdot G^2/2$$

Entweder die Kosten sind fast optimal, oder mit  $x^{t+1}$  kommen wir näher an  $x^*$ .

*Beweis.* Wir stellen fest, dass  $\|P(x)\| \leq \|x\|$ , da  $D$  konvex und die Projektion  $x$  immer in Richtung von  $D$  verschiebt (und damit näher zu  $x^* = 0 \in D$ ).

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= \frac{1}{2\eta} (\|x^{t+1}\|^2 - \|x^t\|^2) \\ &\leq \frac{1}{2\eta} (\|x^t - \eta \nabla c^t(x^t)\|^2 - \|x^t\|^2) \quad (\text{da } \|P(x)\| \leq \|x\|) \\ &= \frac{1}{2\eta} (\|x^t\|^2 + \eta^2 \|\nabla c^t(x^t)\|^2 - 2\eta \nabla c^t(x^t) x^t - \|x^t\|^2) \end{aligned}$$

Der letzte Schritt nutzt den Kosinussatz für Vektoren,  $\|u + v\|^2 = \|u\|^2 + \|v\|^2 + 2u^T v$ . Vereinfachen und Anwendung der Definition von  $G$  ergibt

$$\begin{aligned} \Phi_{t+1} - \Phi_t &\leq \frac{1}{2\eta} (\|x^t\|^2 + \eta^2 \|\nabla c^t(x^t)\|^2 - 2\eta \nabla c^t(x^t) x^t - \|x^t\|^2) \\ &\leq \frac{1}{2} \eta G^2 - \nabla c^t(x^t) \cdot x^t . \end{aligned}$$

Konvexität bedeutet, dass eine Funktion ‘‘superlinear’’ wächst – formal:

$$c^t(0) - c^t(x^t) \geq \nabla c^t(x^t) \cdot (0 - x^t) .$$

Damit ergibt sich

$$\begin{aligned} \Phi_{t+1} - \Phi_t &\leq \eta G^2 / 2 - \nabla c^t(x^t) \cdot x^t \\ &\leq \eta G^2 / 2 + c^t(0) - c^t(x^t) , \end{aligned}$$

und das Lemma ist bewiesen.  $\square$  (Lemma)

Durch Aufsummieren von  $t = 1, \dots, T$  erhalten wir eine Teleskopsumme, und das Lemma ergibt

$$\begin{aligned} \sum_{t=1}^T (c^t(x^t) - c^t(0) + \Phi_{t+1} - \Phi_t) &= \Phi_{T+1} - \Phi_1 + \sum_{t=1}^T (c^t(x^t) - c^t(0)) \\ &\leq T \cdot \eta G^2 / 2 . \end{aligned}$$

Wir erinnern uns an  $x^* = 0$  und nutzen  $\frac{\Delta^2}{2\eta} \geq \Phi_t \geq 0$  und  $\eta = \frac{\Delta}{G\sqrt{T}}$  für die folgende Herleitung:

$$\begin{aligned} R(T) &= \sum_{t=1}^T (c^t(x^t) - c^t(0)) \\ &\leq \Phi_1 - \Phi_{T+1} + \frac{T\eta G^2}{2} \leq \frac{\Delta^2}{2\eta} + \frac{T\eta G^2}{2} \\ &= \frac{\Delta G \sqrt{T}}{2} + \frac{\Delta G \sqrt{T}}{2} . \end{aligned}$$

Der Satz ist gezeigt.  $\square$  (Satz)

Mit GIGA existiert ein No-Regret Algorithmus für unendliche Strategieräume. Damit können wir nun die Frage betrachten, wie sich ein Nash-Gleichgewicht im Bandbreitenspiel (und in allgemeineren Spielen) einstellen kann.

Konvergieren Spieler mit No-Regret Algorithmen zum Nash-Gleichgewicht?  
Ein schnelles Experiment im Beispiel-Spiel von oben ergibt Folgendes:

**Beispiel 4.29.** *Bandbreitenspiel*

| $t$ | Spieler 1             |         | Spieler 2             |         | Spieler 3             |         |
|-----|-----------------------|---------|-----------------------|---------|-----------------------|---------|
|     | $\nabla u_1(s^{t-1})$ | $s_1^t$ | $\nabla u_2(s^{t-1})$ | $s_2^t$ | $\nabla u_3(s^{t-1})$ | $s_3^t$ |
| 1   |                       | 0,01000 |                       | 0,01000 |                       | 0,01000 |
| 2   | (21,22222)            | 1,00000 | (43,44444)            | 1,00000 | (65,66667)            | 1,00000 |
| 3   | (-0,77778)            | 0,45003 | (-0,55556)            | 0,60716 | (-0,33333)            | 0,76430 |
| 4   | (-0,58664)            | 0,11133 | (-0,26800)            | 0,45243 | (-0,04408)            | 0,73885 |
| 5   | (-0,29793)            | 0,01000 | (0,00210)             | 0,45348 | (-0,00324)            | 0,73723 |
| 6   | (-0,17409)            | 0,01000 | (0,03659)             | 0,46985 | (-0,03555)            | 0,72133 |
| 7   | (-0,17441)            | 0,01000 | (0,01375)             | 0,47546 | (-0,00227)            | 0,72040 |
| 8   | (-0,17759)            | 0,01000 | (0,00461)             | 0,47720 | (0,00157)             | 0,72099 |
| 9   | (-0,17917)            | 0,01000 | (0,00154)             | 0,47775 | (0,00128)             | 0,72145 |
| 10  | (-0,17984)            | 0,01000 | (0,00051)             | 0,47792 | (0,00075)             | 0,72170 |
| 11  | (-0,18012)            | 0,01000 | (0,00016)             | 0,47797 | (0,00040)             | 0,72182 |
| 12  | (-0,18024)            | 0,01000 | (0,00004)             | 0,47798 | (0,00021)             | 0,72189 |
| 13  | (-0,18029)            | 0,01000 | (0,00000)             | 0,47798 | (0,00011)             | 0,72192 |
| 14  | (-0,18031)            | 0,01000 | (-0,00001)            | 0,47798 | (0,00006)             | 0,72193 |
| 15  | (-0,18032)            | 0,01000 | (-0,00001)            | 0,47797 | (0,00003)             | 0,72194 |
| 16  | (-0,18033)            | 0,01000 | (-0,00001)            | 0,47797 | (0,00002)             | 0,72195 |
| 17  | (-0,18033)            | 0,01000 | (-0,00001)            | 0,47797 | (0,00001)             | 0,72195 |
| 18  | (-0,18033)            | 0,01000 | (-0,00000)            | 0,47797 | (0,00000)             | 0,72195 |

■

Das Bandbreitenspiel ist ein Spezialfall einer großen Klasse von Spielen, die *sozial konkave Spiele* genannt werden. Jedes sozial konkave Spiel hat ein reines Nash-Gleichgewicht.

**Definition 4.30** (Sozial konkaves Spiel). Ein *sozial konkaves Spiel* ist ein strategisches Spiel  $\Gamma = (\mathcal{N}, (\Sigma_i)_{i \in \mathcal{N}}, (u_i)_{i \in \mathcal{N}})$ .

1.  $\mathcal{N}$  ist eine endliche Menge von  $n$  Spielern.
2. Jede Strategiemenge  $\Sigma_i$  ist kompakt und konvex.
3. Nutzenfunktion  $u_i(s_i, s_{-i})$  ist konkav in  $s_i$  für jedes gegebene  $s_{-i}$ .
4. Nutzenfunktion  $u_i(s_i, s_{-i})$  ist konvex in  $s_{-i}$  für jedes gegebene  $s_i \in \Sigma_i$ .
5. Es gibt  $(\lambda_i)_{i \in \mathcal{N}}$  mit  $\lambda_i > 0$ ,  $\sum_i \lambda_i = 1$ , so dass  $g(s) = \sum_{i \in \mathcal{N}} \lambda_i u_i(s)$  eine konkave Funktion in  $s$  ist.

**Satz 4.31.** Wenn in einem wiederholten, sozial konkaven Spiel über  $T$  Zeitschritte jeder Spieler einen No-Regret Algorithmus für die Strategiewahl nutzt, dann gilt für  $T \rightarrow \infty$ :

1. Der durchschnittliche Ablauf des Spiels  $\hat{s}$  konvergiert zu einem (gemischten) Nash-Gleichgewicht.
2. Der durchschnittliche Nutzen jedes Spielers konvergiert zum Nutzen im gemischten Nash-Gleichgewicht.

Wenn alle Spieler im wiederholten Bandbreitenspiel den GIGA Algorithmus für die Strategiewahl nutzen, dann konvergiert der durchschnittliche Ablauf des Spiels zu einem Nash-Gleichgewicht. Daneben gibt es bei streng konkaven Nutzenfunktionen  $u_i$  ein eindeutiges gemischtes Gleichgewicht – und dies ist auch ein reines Gleichgewicht.

Daher beweist der Satz unsere Intuition aus dem Experiment: Im Bandbreitenspiel können die Spieler mit dem GIGA Algorithmus ein reines Nash-Gleichgewicht erlernen.

## 4.2 Lernen von korrelierten Gleichgewichten

Können wir auch korrelierte Gleichgewichte erlernen?

Dafür betrachten wir Algorithmen mit einer stärkeren Garantie im Experten Problem. Für Algorithmus  $H$  sei  $H(t) \in [N]$  der in Schritt  $t$  gewählte Experte. Bisher haben wir den Verlust des *besten einzelnen Experten* als Vergleichswert für Regret betrachtet. Jetzt betrachten wir die *beste Funktion von Experten*

$$\sigma^* \in \arg \min_{\sigma: [N] \rightarrow [N]} \sum_{t=1}^T \ell_{\sigma(H(t))}^t,$$

mit  $L_{s \min}^T = \sum_{t=1}^T \ell_{\sigma^*(H(t))}^t$  dem Verlust der besten Funktion von Experten.

**Definition 4.32** (Swap-Regret). Der *Swap-Regret* von  $H$  ist gegeben durch

$$SR(T) = L_H^T - L_{s \min}^T = \sum_{t=1}^T \ell_H^t - \sum_{t=1}^T \ell_{\sigma^*(H(t))}^t.$$

Swap-Regret beschreibt den Verlust gegenüber Austausch von Experten. Wenn  $\sigma^*(i) = j$ , dann bedeutet dies:

Wenn immer  $H$  Experte  $i$  gewählt hat, hätte er lieber Experte  $j$  wählen sollen.

**Definition 4.33** (No-Swap-Regret Algorithmus). Für einen No-Swap-Regret Algorithmus gilt  $\frac{SR(T)}{T} \rightarrow 0$  für  $T \rightarrow \infty$ .

Es gilt  $SR(T) \geq R(T)$  und No-Swap-Regret  $\Rightarrow$  No-Regret. (Warum?)

Wenn wir über Swap-Regret reden, kann unser Gegner die Auswahl von Experten verändern, indem er Experte  $i$  immer durch einen anderen Experten  $j$  ersetzt. Dadurch hat unser Gegner deutlich mehr Möglichkeiten den Verlust zu erhöhen als beim Regret.

Betrachte einen Ablauf des Spiels  $s^1, s^2, \dots, s^T$  über  $T$  Runden und interpretiere ihn als Verteilung über Zustände. Wenn Spieler  $i$  einen Swap-Regret von  $SR_i(T)$  hat, dann gilt für jede Strategie  $s_i$  und jede Funktion  $\sigma: S_i \rightarrow S_i$ :

$$\begin{aligned} \mathbb{E}_{k \in [T]} [c_i(s^k)] &= \sum_{t=1}^T \frac{1}{T} \cdot c_i(s_i^t, s_{-i}^t) \\ &\leq \sum_{t=1}^T \frac{1}{T} \cdot c_i(\sigma(s_i^t), s_{-i}^t) + \frac{SR_i(T)}{T} \\ &= \mathbb{E}_{k \in [T]} [c_i(\sigma(s_i^k), s_{-i}^k)] + \frac{SR_i(T)}{T}. \end{aligned}$$

**Proposition 4.34.** Wenn jeder Spieler nach  $T$  Runden Swap-Regret höchstens  $R$  hat, dann stellt der Ablauf des Spiels ein  $\frac{R}{T}$ -approximatives korreliertes Gleichgewicht dar.

Wir können No-Regret Algorithmen in No-Swap-Regret Algorithmen verwandeln! Wir betrachten eine Black-Box-Reduktion.

**Satz 4.35.** Sei  $H$  ein Algorithmus mit Regret  $R(T)$ . Dann existiert ein Algorithmus  $M$  mit Swap-Regret  $SR(T) = N \cdot R(T)$ .

*Beweis.* Wir nutzen  $N$  Kopien  $H_1, \dots, H_N$  von Algorithmus  $H$ . In gewisser Weise sichert uns  $H_j$  zu, dass keine Abbildung  $\sigma$  von Experte  $j$  auf  $\sigma(j)$  allzu großen Swap-Regret erzeugt.

Ein *Master-Algorithmus*  $M$  koordiniert die Kopien von  $H$ .

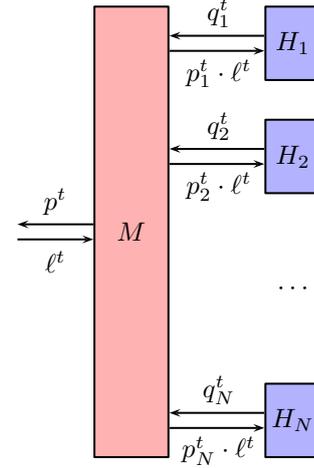
**Algorithmus 6:** Master-Algorithmus

---

**for** jeden Schritt  $t$  **do**

 Erhalte Verteilungen  $q_1^t, \dots, q_n^t$  von  
 $H_1, \dots, H_N$   
 Errechne Konsens-Verteilung  $p^t$   
 Wähle Experte  $i$  mit W.keit  $p_i^t$   
 Melde an  $H_j$  einen Verlust von  $p_j^t \cdot \ell_i^t$ , für  
 jeden Experten  $i \in N$ 


---



Wir betrachten die Errechnung der Konsens-Verteilung  $p^t$  am Ende des Beweises.  
 Für Swap-Regret betrachten wir die folgenden Terme:

- Verlust des Master-Algorithmus:  $L_M^T = \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_i^t$
- Verlust mit bester Abbildung  $\sigma^*$ :  $L_{s \min}^T = \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_{\sigma^*(i)}^t$

Algorithmus  $H_j$  denkt, wir wählen Experte  $i$  mit Wahrscheinlichkeit  $q_{j,i}^t$  und die Kosten dafür wären  $p_j^t \ell_i^t$ .

- Interner Verlust von  $H_j$ :  $L_{H_j}^T = \sum_{t=1}^T \sum_{i \in [N]} q_{j,i}^t \cdot (p_j^t \ell_i^t)$
- Bester Experte  $k_j$  für  $H_j$ :  $L_{\min,j}^T = \sum_{t=1}^T p_{k_j}^t \ell_{k_j}^t$

$H_j$  hat Regret höchstens  $R(T)$ , also  $L_{H_j}^T \leq L_{\min,j}^T + R(T)$ .

Sei nun  $\sigma^*$  die beste Abbildung der Experten. Dann gilt mit Aufsummieren:

$$\begin{aligned}
 \sum_{j \in [N]} L_{H_j}^T &\leq \sum_{j \in [N]} L_{\min,j}^T + R(T) \\
 &\leq \sum_{j \in [N]} \sum_{t=1}^T p_j^t \ell_{\sigma^*(j)}^t + \sum_{j \in [N]} R(T) \\
 &= L_{s \min}^T + N \cdot R(T)
 \end{aligned}$$

Für die Summe der internen Verluste haben wir die No-Swap-Regret Garantie.

Wie hängt der tatsächliche Verlust  $L_M^T$  mit der Summe der internen Verluste  $\sum_j L_{H_j}^T$  zusammen?

Dafür betrachten wir nun die Konstruktion der Konsens-Verteilung  $p^t$ .

Wähle die Konsens-Verteilung  $p^t$  als

$$p_i^t = \sum_{j \in [N]} q_{j,i}^t p_j^t .$$

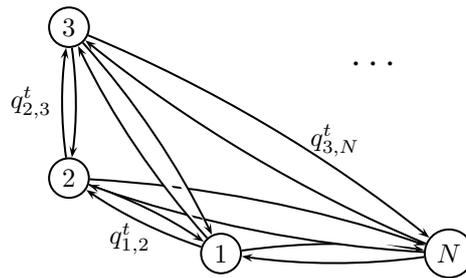
Dann gilt:

$$\begin{aligned}
 L_M^T &= \sum_{t=1}^T \sum_{i \in [N]} p_i^t \ell_i^t \\
 &= \sum_{t=1}^T \sum_{i \in [N]} \sum_{j \in [N]} q_{j,i}^t p_j^t \ell_i^t \\
 &= \sum_{j \in [N]} L_{H_j}^T \\
 &\leq L_{s \min}^T + N \cdot R(T) \text{ ,}
 \end{aligned}$$

und der Satz ist gezeigt. □

$p^t$  ist die *stationäre Verteilung der folgenden Markov-Kette*:  
 Die Zustände sind alle Experten. Wenn wir bei Zustand/Experte  $i$  sind, dann ist die Wahrscheinlichkeit zu Zustand/Experte  $j$  zu wechseln gegeben durch  $q_{i,j}^t$ . Matrix  $Q^t$  mit  $Q^t(i, j) = q_{i,j}^t$  gibt die Übergangswahrscheinlichkeiten von Zustand/Experte  $i$  an.

Wähle  $p^t$  als die stationäre Verteilung der Markov-Kette, der dominante Eigenvektor  $p^t$  der Matrix  $Q^t$  kann effizient berechnet werden.



### 4.3 Zusammenfassung

In diesem Kapitel haben wir Algorithmen kennen gelernt, mit denen die Spieler gute Strategien erlernen können, die ein Gleichgewicht bilden. No-Swap-Regret Algorithmen konvergieren für endliche Spiele in polynomieller Zeit in einem korrelierten Gleichgewicht. Dies ist eine Verallgemeinerung von gemischten Nash-Gleichgewichten und entspricht einer Verteilung über Zustände, die keine profitablen "Swap"-Abweichungen besitzt.

Eine weitere Klasse sind No-Regret Algorithmen, welche für endliche Spiele in polynomieller Zeit in grob-korrelierten Gleichgewichten mit No-Regret konvergieren. Wir nennen ein grob-korreliertes Gleichgewicht eine Verteilung ohne profitable "bester-Experte"-Abweichungen. No-Regret Algorithmen lassen sich ebenfalls auf 2-Spieler-Nullsummenspiele und sozial konkave Spiele anwenden, wobei die Algorithmen in polynomieller Zeit zu gemischten Nash-Gleichgewichten konvergieren. Beachte: Konvergenz für No-Regret oder No-Swap-Regret bezieht sich auf  $\varepsilon$ -approximative Varianten der Gleichgewichte.  $\varepsilon$  sinkt als Funktion von  $T$ .

Zur Vertiefung wird folgende Literatur empfohlen:

- Nisan, Roughgarden, Tardos, Vazirani. Algorithmic Game Theory, 2007. (Kapitel 4).

- Littlestone, Warmuth. The Weighted Majority Algorithm. *Information & Computation* 108(2):212–261, 1994.
- Freund, Schapire. Adaptive Game Playing using Multiplicative Weights. *Games and Economic Behavior*, 29:79–103, 1999.
- Rosen. Existence and Uniqueness of Equilibrium Points for Concave n-Person Games. *Econometrica*, 33(3):520–534, 1965.
- Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. *ICML 2003*.
- Even-Dar, Mansour, Nadav. On the Convergence of Regret Minimization Dynamics in Concave Games. *STOC 2009*.
- Blum, Mansour. From External to Internal Regret. *Journal of Machine Learning Research* 8:1307–1324, 2007.
- Roughgarden. *Twenty Lectures on Algorithmic Game Theory*, 2016. (Kapitel 17+18).

# 5 | Preis der Anarchie und Preis der Stabilität

Bisher haben wir verschiedene Gleichgewichte betrachtet und deren Existenz und Berechenbarkeit analysiert. Zum Beispiel wurde das Gefangenendilemma betrachtet, bei dem der optimale Zustand mit jeweils Kosten 2 für die Spieler im Gleichgewicht niemals gespielt wird. Wie viel schlechter sind die Gleichgewichte als die optimalen Zustände? Wie viel schlechter werden die Gleichgewichte durch weitere Verallgemeinerungen?

Um die Qualität der Gleichgewichte einzugrenzen, betrachten wir zwei Konzepte.

Als untere Schranke betrachten wir den Preis der Anarchie und definieren diesen für reine Nash-Gleichgewichte:

Für ein strategisches Spiel  $\Gamma$  bezeichnen wir  $cost(s)$  als *soziale Kosten* für Zustand  $s$  von  $\Gamma$ . Wir betrachten hier nur  $cost(s) = \sum_{i \in \mathcal{N}} c_i(s)$ . Bezeichne  $\Sigma^{RNG}$  die Menge der reinen Nash-Gleichgewichte von  $\Gamma$ .

**Definition 5.1** (Preis der Anarchie). Der *Preis der Anarchie* (PoA) ist das Verhältnis:

$$PoA = \frac{\max_{s' \in \Sigma^{RNG}} cost(s')}{\min_{s \in \Sigma} cost(s)} .$$

PoA quantifiziert die Verschlechterung des teuersten NG im Vergleich zum optimalen Zustand des Spiels.

Analog zum Preis der Anarchie bildet der Preis der Stabilität eine obere Schranke. Sie gibt an, wie gut ein Gleichgewicht werden kann.

Wieder definieren wir zuerst den Preis der Stabilität für reine NG, betrachte dabei  $\Sigma^{RNG}$  als die Menge der reinen NG des Spiels  $\Gamma$ .

**Definition 5.2** (Preis der Stabilität). Der *Preis der Stabilität* ist das Verhältnis:

$$PoS = \frac{\min_{s' \in \Sigma^{RNG}} cost(s')}{cost(s^*)} .$$

PoS quantifiziert die Verschlechterung des billigsten NG im Vergleich zum optimalen Zustand des Spiels.

Wenn wir den Preis der Stabilität für eine Klasse von Spielen bestimmen, suchen wir nach dem schlechtesten besten Nash-Gleichgewicht innerhalb der Klasse. Im Vergleich dazu ist das schlechteste Gleichgewicht für den Preis der Anarchie erforderlich.

Gibt es eine Technik, mit der sich der Preis der Anarchie in vielen Spielen bestimmen lässt? Zur Beantwortung dieser Frage betrachten zwei Klassen von Spielen.

## 5.1 Wardropspiele

Ein *Wardropspiel* ist gegeben durch

- einen gerichteten Graphen  $G = (V, E)$
- $k$  Arten von Verkehr (genannt “Commodities”) mit Quelle-Senke-Paar  $(s_i, t_i)$  und Aufkommen (genannt “Rate”)  $r_i$ , für  $i \in [k] = \{1, \dots, k\}$
- Latenzfunktionen  $d_e : [0, 1] \rightarrow \mathbb{R}$ , für  $e \in E$
- Commodity  $i$  besteht aus überabzählbar unendlich vielen, infinitesimal kleinen Spielern mit Gesamtmasse  $r_i$ .

Wardropspiele dienen als Modell für Systeme mit einer Vielzahl von Nutzern, in denen der Einfluß eines einzelnen Akteurs sehr gering ist (z.B. riesige Verkehrs- oder Computernetzwerke).

Zur Vereinfachung machen wir folgende Annahmen:  
Alle Latenzfunktionen sind nicht-negativ, nicht-fallend und konvex. Des Weiteren betrachten wir Spiele mit *einer Commodity*, Quelle  $s$  und Senke  $t$ , und normalisieren die Rate auf  $r_1 = 1$ .

Flüsse und Latenzen im Wardropspiel werden bezeichnet:

- Sei  $\mathcal{P}$  die Menge von Pfaden  $P$  von  $s$  nach  $t$ .
- Ein *Fluß* liefert einen Flußwert  $f_P \in [0, 1]$ , für jeden  $P \in \mathcal{P}$ , mit  $\sum_{P \in \mathcal{P}} f_P = 1$ .
- Der *Kantenfluß* auf  $e \in E$  ist  $f_e = \sum_{P \ni e} f_P$ .
- Die *Latenz auf Kante*  $e \in E$  bei Fluß  $f$  ist  $d_e(f_e)$ .
- Die *Latenz auf Pfad*  $P \in \mathcal{P}$  bei Fluß  $f$  ist  $d_P(f) = \sum_{e \in P} d_e(f_e)$ .

**Definition 5.3** (Wardrop-Gleichgewicht). Ein Fluß  $f$  ist ein *Wardrop-Gleichgewicht*, wenn für jedes Paar von Pfaden  $P_1, P_2 \in \mathcal{P}$  mit  $f_{P_1} > 0$  gilt  $d_{P_1}(f) \leq d_{P_2}(f)$ .

**Beobachtung:** Im Wardrop-Gleichgewicht werden nur Flusswerte angegeben, aber kein expliziter  $s$ - $t$ -Pfad für jeden der unendlich vielen einzelnen Spieler. Wenn wir die Spieler gemäß der Flusswerte auf Pfade verteilen, bekommt jeder Spieler einen Pfad mit momentan optimalen Kosten. Im Wardrop-Gleichgewicht könnte es aber auch abzählbar unendlich viele Spieler geben, die suboptimale Pfade wählen. Da diese infinitesimal klein sind, fallen sie bei der Flußbedingung nicht ins Gewicht.

Als Optimalitätskriterium betrachten wir soziale Kosten. Bei Wardropspielen entsprechen diese der durchschnittlichen Latenz aller Pfade.

**Definition 5.4** (Soziale Kosten). Die *sozialen Kosten* eines Flusses  $f$  sind das gewichtete Mittel der Spieler/Pfadlatenzen

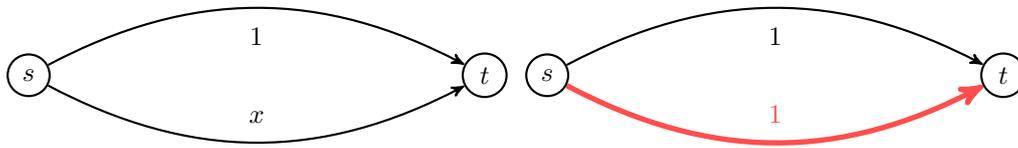
$$\text{cost}(f) = \sum_{P \in \mathcal{P}} d_P(f) \cdot f_P = \sum_{e \in E} d_e(f_e) f_e .$$

**Beispiel 5.5.** *Pigou-Beispiel*

Alle Spieler starten an der Quelle  $s$  und suchen nach einem möglichst günstigen Pfad zur Senke  $t$ .

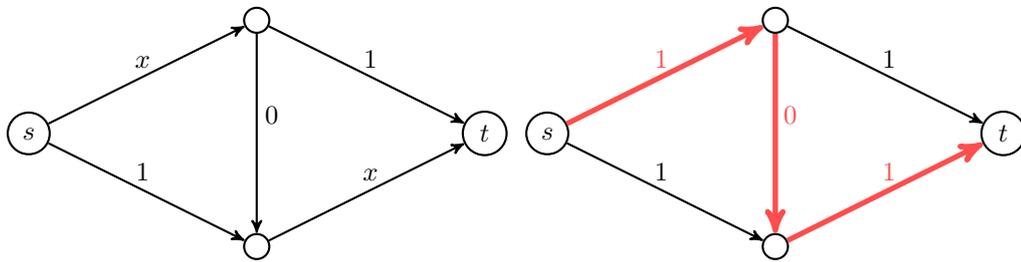
Stellen wir uns vor, ein Spieler nach dem anderen würde sich für eine der beiden Kanten entscheiden. Jeder (bis auf höchstens abzählbar unendlich viele) der überabzählbar unendlich vielen

Spieler entscheidet sich für die untere Kante, deren Kosten erst 1 betragen, wenn sie von allen Spielern (bis auf abzählbar unendlich vielen) gewählt wurde.

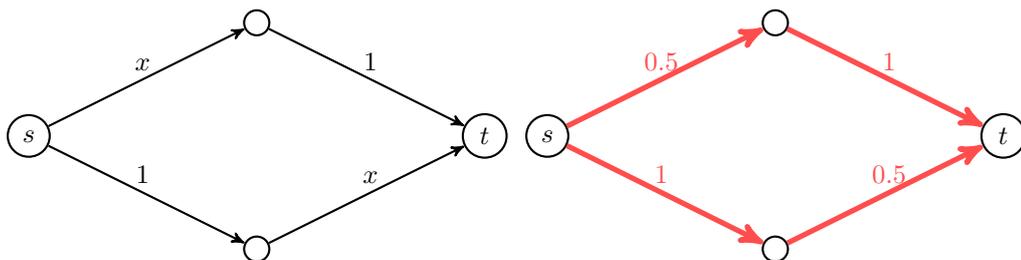


In diesem Beispiel betragen die sozialen Kosten 1. (Gibt es einen besseren Fluss?)

Das nächste Beispiel besitzt mehr mögliche Pfade für die Spieler. Auch hier werden diese den Pfad mit der geringsten Latenz wählen. Analog bevorzugen die Spieler die Kanten mit Latenzfunktion  $d_e(x) = x$  und  $d_e(x) = 0$ .



Die Latenz des gegebenen Flusses beträgt 2, dabei können die sozialen Kosten 1.5 erreicht werden. Wir betrachten das Beispiel erneut in leicht veränderter Version. Ohne die Kante mit Latenz 0 sind die beiden verbleibenden Pfade gleich gut. Daher wird sich der Fluß gleichmäßig auf beide Pfade verteilen.



Die Latenz beträgt wie die sozialen Kosten 1.5. Durch das Zerstören einer schnellen Verbindung *verbessern* sich die Kosten im Gleichgewicht *für jeden Spieler*. Dieses Phänomen nennt sich Braess Paradoxon. ■

Wie gut sind die sozialen Kosten in einem Wardrop-Gleichgewicht?  
Welche Parameter des Spiels haben Einfluß auf die sozialen Kosten?

**Satz 5.6** (Roughgarden, Tardos, 2002). *Der Preis der Anarchie in Wardropspielen mit affinen Latenzfunktionen ist höchstens  $4/3$ .*

*Beweis.* Sei  $f$  ein Fluss im Gleichgewicht. Sei  $g$  ein beliebiger Fluss. Aufgrund der Gleichgewichts-

bedingung gilt

$$\begin{aligned} \text{cost}(f) &= \sum_{P \in \mathcal{P}} f_P d_P(f) \\ &\leq \sum_{P \in \mathcal{P}} g_P d_P(f) \end{aligned}$$

**Lemma 5.7.** *Es gilt*

$$\sum_{P \in \mathcal{P}} g_P d_P(f) \leq 1 \cdot \text{cost}(g) + \frac{1}{4} \cdot \text{cost}(f) .$$

Anwendung des Lemmas liefert  $\text{cost}(f) \leq \text{cost}(g) + \text{cost}(f)/4$ , woraus folgt  $\frac{3}{4}\text{cost}(f) \leq \text{cost}(g)$  für jeden Fluß  $g$ . Der Satz ist gezeigt.  $\square$  (Satz)

*Beweis (Lemma 5.7).*

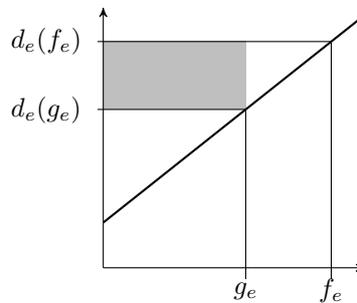
$$\begin{aligned} \sum_{P \in \mathcal{P}} g_P d_P(f) &= \sum_{e \in E} g_e d_e(f_e) \\ &= \sum_{e \in E} g_e (d_e(g_e) + d_e(f_e) - d_e(g_e)) \\ &= C(g) + \sum_{e \in E} g_e (d_e(f_e) - d_e(g_e)) . \end{aligned}$$

Wir zeigen nun für alle  $e$

$$g_e (d_e(f_e) - d_e(g_e)) \leq \frac{1}{4} f_e d_e(f_e) .$$

Das gilt offensichtlich, wenn  $f_e < g_e$ , da in diesem Fall  $d_e(f_e) \leq d_e(g_e)$ . Damit ist die linke Seite der Gleichung höchstens 0.

Sei nun  $f_e \geq g_e$  wie im folgenden Bild der Funktion  $d_e$ .



Aus dem Bild ist erkennbar, dass die graue Fläche immer höchstens  $1/4$  der Gesamtfläche des äußeren Rechtecks beträgt. Dadurch ergibt sich

$$\begin{aligned} \sum_{p \in \mathcal{P}} g_P d_P(f) &= \text{cost}(g) + \sum_{e \in E} g_e (d_e(f) - d_e(g)) \\ &\leq \text{cost}(g) + \frac{1}{4} \sum_{e \in E} f_e d_e(f) \\ &= \text{cost}(g) + \frac{1}{4} \text{cost}(f) . \end{aligned}$$

$\square$  (Lemma)

**Theorem 5.8** (Roughgarden, Tardos, 2002). *In Wardropspielen, in denen die Latenzfunktionen positive Polynome vom Grad  $d$  sind, ist der Preis der Anarchie ist höchstens  $d + 1$ .*

### 5.1.1 Existenz und Eindeutigkeit von Wardrop-Gleichgewichten

**Satz 5.9** (Beckmann, McGuire, Winsten, 1956). *In jedem Wardropspiel mit stetigen Latenzfunktionen gibt es mindestens ein Wardrop-Gleichgewicht. Für stetige, streng monoton steigende Funktionen sind die Kantenflüsse im Gleichgewicht eindeutig.*

*Beweis.* Für den Beweis wird die *Potenzialfunktion* des Wardropspiels verwendet:

$$\Phi(f) = \sum_{e \in E} \int_{x=0}^{f_e} d_e(x) dx .$$

Wir betrachten einen Fluß, der das Potenzial minimiert, und formulieren dies als Optimierungsproblem:

$$\begin{aligned} \text{Minimiere} \quad & \sum_{e \in E} \int_{x=0}^{f_e} d_e(x) dx \\ \text{so dass} \quad & f_e = \sum_{P \in \mathcal{P}} f_P \quad \text{für jedes } e \in E \\ & \sum_{P \in \mathcal{P}} f_P = 1 \\ & f_P \geq 0 \quad \text{für jedes } P \in \mathcal{P} \end{aligned} \tag{5.1}$$

Jede Latenzfunktion  $d_e$  ist stetig. Daher ist

$$D_e(f_e) = \int_{x=0}^{f_e} d_e(x) dx$$

stetig differenzierbar. Somit existiert eine optimale Lösung des Problems. Mit  $f_e = \sum_{P \in \mathcal{P}} f_P$  gilt für die Ableitung

$$\frac{d}{df_P} \sum_{e \in E} \int_{x=0}^{f_e} d_e(x) dx = \sum_{e \in P} d_e(f_e) = d_P(f_P) .$$

Die Rate 1 muss optimal auf die Pfade verteilt werden. Wir verringern  $\Phi(f)$ , indem wir  $f_P$  mit hoher Ableitung verringern und dafür  $f_{P'}$  mit kleiner Ableitung erhöhen. Im Optimum kann also kein Fluß von Pfaden mit höherer Latenz auf Pfade mit kleinerer Latenz verschoben werden.

Daher gilt in der optimalen Lösung für jeden Pfad  $P$  mit  $f_P > 0$  und jeden beliebigen Pfad  $P'$ , dass  $d_P(f) \leq d_{P'}(f)$ . Die optimale Lösung ist also ein Wardrop-Gleichgewicht.

Wenn  $d_e$  streng monoton steigend in  $f_e$  ist, gilt das auch für  $d_P$  in  $f_P$ . Dann ist  $\Phi(f)$  streng konvex in  $f_P$ . Wenn  $\Phi(f)$  streng konvex ist, dann ist die optimale Lösung für die  $f_e$  eindeutig.  $\square$

Anmerkungen:

- Eindeutige Kantenflüsse  $\Rightarrow$  Eindeutige soziale Kosten im Gleichgewicht. Daher gilt in diesen Spielen: Preis der Anarchie = Preis der Stabilität.
- Eindeutigkeit gilt nicht für die Aufteilung in Pfadflüsse  $f_P$ .
- Problem (5.1) kann mit dem Ellipsoidverfahren für konvexe Minimierungsprobleme in polynomieller Zeit gelöst werden (approximativ bzgl. einer vorher bestimmten numerischen Genauigkeit).

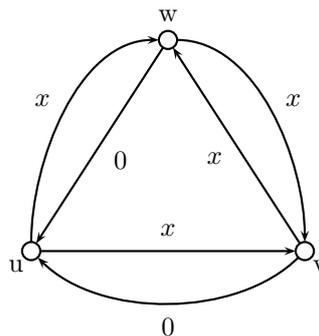
**Korollar 5.10.** *Ein Wardrop-Gleichgewicht kann in polynomieller Zeit berechnet werden.*

## 5.2 Auslastungsspiele

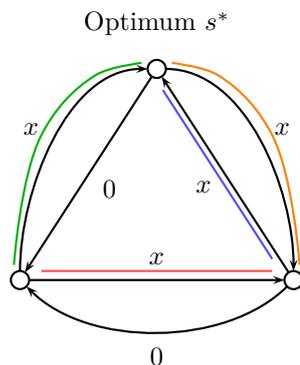
Wir betrachten den Preis der Anarchie in Auslastungsspielen mit linearen Verzögerungsfunktionen  $d_r(x) = a_r \cdot x$ , für  $a_r \geq 0$ .

**Beispiel 5.11.** *Auslastungsspiel mit linearen Verzögerungsfunktionen*

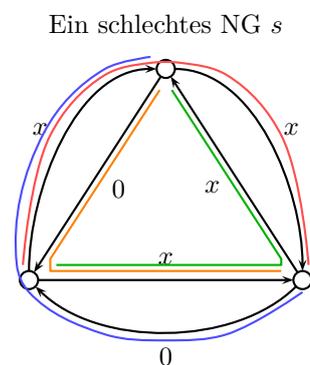
Im folgenden Spiel gibt es 4 Spieler. Die Spieler möchten von (1)  $u$  nach  $w$ , (2)  $w$  nach  $v$ , (3)  $v$  nach  $u$  und (4)  $u$  nach  $v$ . Im Prinzip hat jeder Spieler nur eine direkte (direkte Kante) und eine indirekte (über dritten Knoten) Strategie:



In diesem Spiel vergleichen wir das Optimum mit dem schlechtesten reinen Nash-Gleichgewicht.



$$\text{cost}(s^*) = 1 + 1 + 1 + 1 = 4$$



$$\text{cost}(s) = 3 + 2 + 3 = 10$$

Der PoA in in diesem Spiel ist mindestens 2.5. Geht es noch schlechter? ■

**Satz 5.12.** *Der Preis der Anarchie in Auslastungsspielen mit linearen Verzögerungsfunktionen ist höchstens 2.5.*

*Beweis.* Sei  $s$  das schlechteste reine Nash-Gleichgewicht. Wenn Spieler  $i$  in  $s$  zu einer anderen Strategie abweicht, werden seine Kosten nicht kleiner. Sei  $s^*$  ein optimaler Zustand. Es gilt insbesondere, dass  $c_i(s) \leq c_i(s_i^*, s_{-i})$ .

Damit können wir die sozialen Kosten beschränken durch

$$\text{cost}(s) = \sum_{i \in \mathcal{N}} c_i(s) \leq \sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) . \quad (5.2)$$

Dies ist eine verschränkte Summe – jeder Spieler betrachtet die Kosten, wenn er alleine zur Strategie in  $s^*$  abweichen würde. Wie können wir diesen Term auf  $\text{cost}(s)$  und  $\text{cost}(s^*)$  zurückführen?

Betrachten wir den Term genauer für die Auslastungsspiele. Wir schreiben  $n_r = n_r(s)$  für die Last von Ressource  $r$  in Zustand  $s$  und  $n_r^* = n_r(s^*)$  für  $s^*$ .

Wenn Spieler  $i$  nach  $s_i^*$  abweicht, dann sieht er auf jeder Ressource  $r \in s_i^*$  eine Last von höchstens  $n_r + 1$  (evtl. nur  $n_r$ , wenn  $r \in s_i \cap s_i^*$ )

$$\sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) \leq \sum_{i \in \mathcal{N}} \sum_{r \in s_i^*} d_r(n_r + 1) .$$

Da genau  $n_r^*$  Spieler auf Ressource  $r$  abweichen, gilt:

$$\sum_{i \in \mathcal{N}} \sum_{r \in s_i^*} d_r(n_r + 1) = \sum_{r \in \mathcal{R}} n_r^* d_r(n_r + 1) = \sum_{r \in \mathcal{R}} n_r^* a_r \cdot (n_r + 1) .$$

Wir nutzen nun folgendes Lemma ohne Beweis:

**Lemma 5.13** (Christodoulou, Koutsoupias, 2005). *Für alle nicht-negativen ganzen Zahlen  $y, z \in \{0, 1, 2, 3, \dots\}$  gilt*

$$y(z + 1) \leq \frac{5}{3} \cdot y^2 + \frac{1}{3} \cdot z^2 .$$

Mit  $y = n_r^*$  und  $z = n_r$  gilt also:

$$\begin{aligned} \sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) &\leq \sum_{r \in \mathcal{R}} a_r n_r^* (n_r + 1) \\ &\leq \sum_{r \in \mathcal{R}} a_r \left( \frac{5}{3} (n_r^*)^2 + \frac{1}{3} n_r^2 \right) \\ &= \frac{5}{3} \sum_{r \in \mathcal{R}} n_r^* (a_r n_r^*) + \frac{1}{3} \sum_{r \in \mathcal{R}} n_r (a_r n_r) \\ &= \frac{5}{3} \cdot \text{cost}(s^*) + \frac{1}{3} \cdot \text{cost}(s) \end{aligned}$$

und somit

$$\sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) \leq \frac{5}{3} \cdot \text{cost}(s^*) + \frac{1}{3} \cdot \text{cost}(s) . \quad (5.3)$$

Damit können wir nun den Preis der Anarchie wir folgt beschränken:

$$\begin{aligned} \text{cost}(s) &\leq \frac{5}{3} \cdot \text{cost}(s^*) + \frac{1}{3} \cdot \text{cost}(s) \\ \Rightarrow \text{PoA} = \frac{\text{cost}(s)}{\text{cost}(s^*)} &\leq \frac{5/3}{1 - 1/3} = 2.5 . \end{aligned} \quad (5.4)$$

□

## 5.3 Smoothness

Aus dem vorangegangenen Beweis leiten wir ein Schema ab:

1. Stelle Ungleichung (5.2) auf. Sie beruht nur auf der Annahme von  $\text{cost}(s)$  und darauf, dass  $s$  ein reines NG ist.
2. Leite Ungleichung (5.3) mit Zahlen  $(\lambda, \mu)$  her. Die Zahlen  $(5/3, 1/3)$  waren hier spezifisch für das Spiel.
3. Schließe die Rechnung durch Ungleichung (5.4) ab. Die Schranke auf den PoA beruht nur auf den Zahlen in Ungleichung (5.3).

Die einzige Information über das Spiel sind die Zahlen in (5.3). Wenn für ein Spiel also eine Ungleichung dieser Form gilt, dann können wir eine Schranke auf den PoA für reine NG mit dem Schema finden.

Lemma 5.7 zeigt eine Ungleichung der Form (5.3).

**Definition 5.14.** Ein Spiel heißt  $(\lambda, \mu)$ -smooth für  $\lambda > 0$  und  $\mu \leq 1$ , wenn für jedes Paar von Zuständen  $s, s'$  gilt

$$\sum_{i \in \mathcal{N}} c_i(s'_i, s_{-i}) \leq \lambda \cdot \text{cost}(s') + \mu \cdot \text{cost}(s) . \quad (5.5)$$

Daraus folgt mit dem Schema dann direkt:

**Satz 5.15.** Wenn ein Spiel  $(\lambda, \mu)$ -smooth ist, dann ist der Preis der Anarchie für reine Nash-Gleichgewichte höchstens

$$\frac{\lambda}{1 - \mu} .$$

Viele Spiele haben keine reinen Nash-Gleichgewichte. Was ist der Preis der Anarchie für allgemeinere Gleichgewichte?

Wie hoch sind die sozialen Kosten, wenn jeder Spieler mit einem No-Regret-Algorithmus spielt?

**Definition 5.16.** Der Preis der Anarchie für grob-korrelierte Gleichgewichte oder Preis der totalen Anarchie ist die kleinste Zahl  $\rho \geq 1$ , so dass für jedes grob-korrelierte Gleichgewicht  $\mathcal{V}$  und für jeden Zustand  $s'$  des Spiels gilt

$$\mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)] \leq \rho \cdot \text{cost}(s') .$$

Im Schema oben haben wir nur benutzt, dass (5.3) für ein reines Nash-Gleichgewicht  $s$  und ein Optimum  $s^*$  gilt. In der Definition fordern wir allerdings, dass (5.5) für jedes Paar von Zuständen gilt. Diese stärkere Forderung erlaubt folgendes Resultat:

**Satz 5.17.** Wenn ein Spiel  $(\lambda, \mu)$ -smooth ist, dann ist der PoA für grob-korrelierte Gleichgewichte höchstens

$$\frac{\lambda}{1 - \mu} .$$

*Beweis.* Sei  $s^*$  ein optimaler Zustand. Wir erhalten Schritt (5.2) wie folgt:

$$\begin{aligned} \mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)] &= \mathbb{E}_{s \sim \mathcal{V}} \left[ \sum_{i \in \mathcal{N}} c_i(s) \right] = \sum_{i \in \mathcal{N}} \mathbb{E}_{s \sim \mathcal{V}} [c_i(s)] \\ &\leq \sum_{i \in \mathcal{N}} \mathbb{E}_{s \sim \mathcal{V}} [c_i(s_i^*, s_{-i})] = \mathbb{E}_{s \sim \mathcal{V}} \left[ \sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) \right] \end{aligned}$$

Nun nutzen wir die Smoothness-Eigenschaft punktweise für alle möglichen Zustände unter der Verteilung  $\mathcal{V}$  und  $s^*$

$$\begin{aligned} \mathbb{E}_{s \sim \mathcal{V}} \left[ \sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) \right] &\leq \mathbb{E}_{s \sim \mathcal{V}} [\lambda \cdot \text{cost}(s^*) + \mu \cdot \text{cost}(s)] \\ &\leq \lambda \cdot \text{cost}(s^*) + \mu \cdot \mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)] . \end{aligned}$$

Dies entspricht Schritt (5.3).

Die verbleibende Herleitung ist nun exakt wie in (5.4):

$$\begin{aligned} \mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)] &\leq \lambda \text{cost}(s^*) + \mu \cdot \mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)] \\ \Rightarrow \frac{\mathbb{E}_{s \sim \mathcal{V}} [\text{cost}(s)]}{\text{cost}(s^*)} &\leq \frac{\lambda}{1 - \mu} . \end{aligned}$$

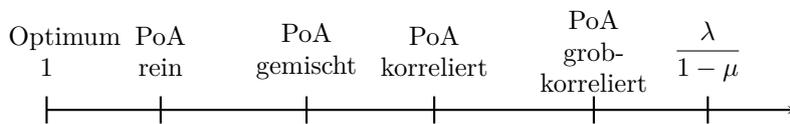
□

**Satz 5.18.** *Jedes Wardropspiel mit affinen Verzögerungsfunktionen ist  $(1, \frac{1}{4})$ -smooth. Damit ist der PoA für grob-korrelierte Gleichgewichte höchstens  $4/3$ .*

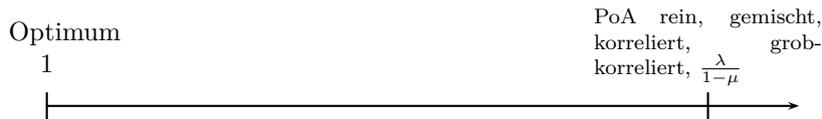
**Satz 5.19.** *Jedes Auslastungsspiel mit linearen Verzögerungsfunktionen ist  $(\frac{5}{3}, \frac{1}{3})$ -smooth. Damit ist der PoA für grob-korrelierte Gleichgewichte höchstens 2.5.*

In beiden Fällen gelten die oberen Schranken für grob-korrelierte Gleichgewichte. Daneben gibt es Instanzen, in denen die Schranke schon für reine Gleichgewichte angenommen wird. Wir nennen Klassen von Spielen mit dieser Eigenschaft dicht (engl.: *tight*).

Smoothness ergibt eine PoA-Schranke für viele Gleichgewichte. Jedes reine Nash-Gleichgewicht ist ein gemischtes Nash-Gleichgewicht, jedes gemischte Nash-Gleichgewicht ein korreliertes Gleichgewicht und jedes korrelierte ein grob-korreliertes Gleichgewicht. Je allgemeiner das Konzept, desto größer der Preis der Anarchie (da worst-case). Die Schranke  $\lambda/(1 - \mu)$  mit Smoothness kann man noch allgemeiner anwenden als nur auf grob-korrelierte Gleichgewichte:



Für *dichte* Klassen von Spielen gibt es ein reines Nash-Gleichgewicht, das einen PoA von  $\lambda/(1 - \mu)$  liefert. Damit kollabiert die Hierarchie:



**Satz 5.20** (Roughgarden, 2003, Informell). *Für eine große Klasse von nicht-fallenden, nicht-negativen Verzögerungsfunktionen ist der PoA für reine Nash-Gleichgewichte in Wardropspielen  $\lambda/(1 - \mu)$ . Diese Schranke wird erreicht in Netzwerken mit zwei Knoten und zwei Links (wie im Pigou-Beispiel).*

**Satz 5.21** (Roughgarden, 2009, Informell). *Für eine große Klasse von nicht-fallenden, nicht-negativen Verzögerungsfunktionen ist der PoA für reine Nash-Gleichgewichte in Auslastungsspielen  $\lambda/(1 - \mu)$ . Diese Schranke wird erreicht in Netzwerken bestehend aus zwei Kreisen und evtl. vielen Knoten (wie im Beispiel für lineare Funktionen oben).*

Viele Klassen von Wardrop und Auslastungsspielen sind also dicht und erlauben universelle Worst-Case-Strukturen der Netzwerke.

### 5.3.1 Grenzen von Smoothness

Smoothness gibt sehr gute Schranken, wenn gute soziale Kosten im Spiel nicht zu stark von koordiniertem Verhalten der Spieler abhängig sind.

Betrachte dagegen folgendes Spiel, in dem niedrige soziale Kosten stark von der Koordination der Spielerstrategien abhängen:

|   |     |     |
|---|-----|-----|
|   | A   | B   |
| A | 1   | 100 |
| B | 100 | 1   |

Die optimalen Zustände sind reine NG mit sozialen Kosten 2, daher beträgt der Preis der Anarchie für reine NG 1.

Das schlechteste gemischte Nash-Gleichgewicht ist  $x_1 = x_2 = (0.5, 0.5)$ . Hier liegen die erwarteten sozialen Kosten bei 101. Dadurch resultiert der Preis der Anarchie von 50.5.

Jedes grob-korrelierte Gleichgewicht  $\mathcal{V}$  ist ein korreliertes Gleichgewicht. Aus den Gleichgewichtsbedingungen folgt:

$$\begin{aligned} & \max(\Pr[s = (A, B)], \Pr[s = (B, A)]) \\ & \leq \min(\Pr[s = (A, A)], \Pr[s = (B, B)]) . \end{aligned}$$

Das schlechteste grob-korrelierte Gleichgewicht ist die Gleichverteilung (also wieder das schlechteste gemischte NG). Analog beträgt der Preis der Anarchie für korrelierte und grob-korrelierte Gleichgewichte also 50.5.

Für eine Schranke auf  $\lambda$  und  $\mu$  betrachte nun das reine Nash-Gleichgewicht  $s = (A, A)$  und ein Optimum  $s^* = (B, B)$ :

$$200 = c_1(B, A) + c_2(A, B) = \sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i}) \leq \lambda \cdot \text{cost}(s^*) + \mu \cdot \text{cost}(s) = 2\lambda + 2\mu .$$

Dann ist  $\lambda \geq 100 - \mu$  und  $PoA \leq (100 - \mu)(1 - \mu) = 1 + 99/(1 - \mu)$ . Mit  $\mu = 0$  ergibt sich also 100 als kleinste obere Schranke. Selbst wenn wir nur die Smoothness-Bedingung bzgl. reiner Nash-Gleichgewichte und optimaler Zustände betrachten, bekommen wir eine Schranke von 100 auf den Preis der Anarchie. Das ist *100 Mal größer* als der echte Preis der Anarchie für reine Nash-Gleichgewichte. Es ist auch fast  $|\mathcal{N}|$  *Mal größer* als der PoA für grob-korrelierte Gleichgewichte. In diesem Spiel braucht es *Koordination für Gleichgewichte und niedrige Kosten*. In  $\sum_{i \in \mathcal{N}} c_i(s_i^*, s_{-i})$  treten dagegen nur Zustände auf mit *Kosten, die extrem viel höher sind* als im Gleichgewicht oder im Optimum. Dadurch muss  $\lambda$  *extrem erhöht* werden und die resultierende Schranke verliert ihre Aussagekraft.

## 5.4 Preis der Stabilität

Wenn wir den Preis der Stabilität für eine Klasse von Spielen bestimmen, suchen wir nach dem schlechtesten besten Nash-Gleichgewicht innerhalb der Klasse. Daher gibt es für die Bestimmung des Preis der Stabilität kein vergleichbar allgemeines Schema. Im Folgenden betrachten wir eine spezielle Klasse von Spielen, die die Analyse vereinfacht.

### 5.4.1 Equal-Sharing-Spiele

Ein Equal-Sharing-Spiel besitzt die Menge  $\mathcal{N}$  von  $n$  Spielern und die Menge  $\mathcal{R}$  von  $m$  Ressourcen. Der Spieler  $i$  wählt eine Teilmenge der Ressourcen, d.h. Strategiemenge  $\Sigma_i \subseteq 2^{\mathcal{R}}$ . Die Ressource  $r \in \mathcal{R}$  hat feste Kosten  $c_r \geq 0$ , welche zu gleichen Teilen den Spielern zugewiesen werden, die  $r$  nutzen.

Equal-Sharing-Spiele sind *Auslastungsspiele mit Latenzen*  $d_r(x) = c_r/x$ .

Soziale Kosten ergeben sich als Summe der Kosten der von mind. einem Spieler benutzten Ressourcen:

$$\text{cost}(s) = \sum_{i \in \mathcal{N}} c_i(s) = \sum_{i \in \mathcal{N}} \sum_{r \in s_i} d_r(n_r) = \sum_{\substack{r \in \mathcal{R} \\ n_r \geq 1}} n_r \cdot c_r / n_r = \sum_{\substack{r \in \mathcal{R} \\ n_r \geq 1}} c_r .$$

**Satz 5.22.** *Jedes Equal-Sharing-Spiel ist  $(n, 0)$ -smooth, der Preis der Anarchie für grob-korrelierte Gleichgewichte also höchstens  $n$ . Die Klasse der Equal-Sharing-Spiele ist dicht, d.h. es gibt Spiele, in denen der Preis der Anarchie für reine Gleichgewichte genau  $n$  ist.*

**Satz 5.23.** *Der Preis der Stabilität für reine Nash-Gleichgewichte in Equal-Sharing-Spielen ist höchstens  $\mathcal{H}_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = O(\log n)$ .*

*Beweis.* Wir verwenden Rosenthals Potential für das Equal-Sharing-Spiel:

$$\begin{aligned}\Phi(s) &= \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r} c_r/i = \sum_{\substack{r \in \mathcal{R} \\ n_r \geq 1}} c_r \cdot \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n_r}\right) \\ &\leq \sum_{\substack{r \in \mathcal{R} \\ n_r \geq 1}} c_r \cdot \mathcal{H}_n \\ &= \text{cost}(s) \cdot \mathcal{H}_n .\end{aligned}$$

In  $\Phi(s)$  verrechnen wir für jeden Spieler, der Ressource  $r$  wählt, einen Beitrag von  $c_r/i$  für ein  $i = 1, \dots, n_r$ . In seinen Kosten  $c_i(s)$  verrechnen wir nur  $c_r/n_r$ . Also gilt für jeden Zustand  $s$  im Spiel:

$$\text{cost}(s) \leq \Phi(s) \leq \text{cost}(s) \cdot \mathcal{H}_n .$$

Wir betrachten nun eine Verbesserungsfolge, die im optimalen Zustand  $s^*$  beginnt. Die Folge endet in einem reinen Nash-Gleichgewicht. Jeder Schritt verringert das Potenzial. Das resultierende Nash-Gleichgewicht  $s$  erfüllt daher  $\Phi(s) \leq \Phi(s^*)$ . Also:

$$\text{cost}(s) \leq \Phi(s) \leq \Phi(s^*) \leq \text{cost}(s^*) \cdot \mathcal{H}_n .$$

Es gibt also ein reines Nash-Gleichgewicht, in dem die sozialen Kosten höchstens um einen Faktor  $\mathcal{H}_n$  größer sind als im Optimum  $s^*$ .  $\square$

### 5.4.2 Potenzialspele

Der Beweis für Equal-Sharing-Spiele ist ein Spezialfall der sog. *Potenzialfunktionsmethode*, einer allgemeinen Technik zur Abschätzung des PoS für Potenzialspele. Seien  $A, B \geq 0$  zwei Zahlen, so dass gilt

$$\text{cost}(s) \leq A \cdot \Phi(s) \quad \forall s$$

und

$$\Phi(s) \leq B \cdot \text{cost}(s) \quad \forall s.$$

Wenn wir nun im optimalen Zustand  $s^*$  eine Verbesserungsfolge starten, erreichen wir ein reines Nash-Gleichgewicht mit

$$\text{cost}(s) \leq A \cdot \Phi(s) \leq A \cdot \Phi(s^*) \leq A \cdot B \cdot \text{cost}(s^*) .$$

Im obigen Fall war  $A = 1$  und  $B = \mathcal{H}_n$ .

**Korollar 5.24.** *Der Preis der Stabilität für reine Nash-Gleichgewichte in ordinalen Potenzialspielen mit Parametern  $A$  und  $B$  ist höchstens  $A \cdot B$ .*

## 5.5 Zusammenfassung

Der Preis der Anarchie und der Preis der Stabilität sind jeweils eine Quantifizierung des teuersten bzw. billigsten Nash-Gleichgewichts im Vergleich zum optimalen Zustand des Spiels. Die Smoothness-Eigenschaft eines Spiels resultiert in einer oberen Schranke für den Preis der Anarchie aller uns bekannten Gleichgewichte. Die Schranke kann durch das beschriebene Schema hergeleitet werden. Mit der Potenzialfunktionsmethode kann eine obere Schranke an den Preis der Stabilität in beliebigen Potenzialspielen ermittelt werden.

Zur Vertiefung wird folgende Literatur empfohlen:

- M. Beckmann, B. McGuire, C. Winsten. *Studies in the Economies of Transportation*. Yale University Press, 1956.

- D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung* 12:258–268, 1968.
- J. Correa, A. Schulz, N. Stier Moses, A Geometric Approach to the Price of Anarchy in Nonatomic Congestion Games. *Games and Economic Behavior* 64(2):457–469, 2008.
- T. Roughgarden, É. Tardos. How Bad is Selfish Routing? *Journal of the ACM* 49(2): 236–259, 2002.
- Nisan, Roughgarden, Tardos, Vazirani. *Algorithmic Game Theory*, 2007. (Kapitel 18 und 19.3).
- Awerbuch, Azar, Epstein. The Price of Routing Unsplittable Flow. *SIAM Journal on Computing* 42(1):160–177, 2013.
- Christodoulou, Koutsoupias. The Price of Anarchy of Finite Congestion Games. *STOC* 2005.
- Blum, Hajiaghayi, Ligett, Roth. Regret Minimization and the Price of Total Anarchy. *STOC* 2008.
- Roughgarden. Intrinsic Robustness of the Price of Anarchy. *Journal of the ACM* 62(5):32:1–32:42, 2015.
- Roughgarden. *Twenty Lectures on Algorithmic Game Theory*, 2016. (Kapitel 14)

# 6 | Entwurf anreizkompatibler Mechanismen

Eine häufige Anwendung der Spieltheorie sind Auktionen, darunter fällt z.B. online angezeigte Werbung. Bei einer Auktion veröffentlichen die Bieter ein Gebot und können dadurch die angebotene Ware kaufen. Eine Vorgehensweise des Auktionators bei einer Auktion nennen wir Mechanismus.

Meist möchte der Auktionator seine Auktion so gestalten, dass jeder Bieter wahrheitsgemäß sagt, was ihm die Ware wert ist. Wie entwirft man solch eine Auktion? Können wir ein Konzept ähnlich einem Gleichgewicht entwickeln, von dem niemand abweichen möchte?

In einem Mechanismus gibt es die Menge  $A$  von möglichen *Ergebnissen*. Das Ziel ist es ein Ergebnis  $a \in A$  zu wählen.

Die Bieter sind Spieler und haben quantifizierbare Präferenzen über die Ergebnisse. Dabei ermöglicht eine gemeinsame Währung einen Nutzentransfer zwischen den Spielern. Die Präferenz von Spieler  $i$  wird beschrieben durch eine *Bewertungsfunktion*  $v_i : A \rightarrow \mathbb{R}$  aus einer bekannten Menge von Funktionen  $V_i \subseteq \mathbb{R}^A$ , wobei  $v_i$  eine *private Information* von Spieler  $i$  ist.

Ein Mechanismus zur Wahl eines Ergebnisses:

1. Frage die Bewertungen aller Spieler ab (*direkte Offenlegung*)
2. Bestimme ein Ergebnis  $a \in A$
3. Bestimme *Zahlungen*  $m_i$  für jeden Spieler  $i$

Der *Nutzen* oder *Utility* von Spieler  $i$  ist  $v_i(a) - m_i$ . Der Nutzen ist *quasi-linear*.

## 6.1 Vickrey-Auktion und Vickrey-Clarke-Groves Mechanismen

### Beispiel 6.1. *Scaled-Bid Auktion*

Ein einzelner Gegenstand wird an einen Bieter verkauft. Die Tabelle zeigt an, wie viel jedem Bieter die Ware wert ist.

|        |   |   |    |    |    |
|--------|---|---|----|----|----|
| Bieter | 1 | 2 | 3  | 4  | 5  |
| Wert   | 9 | 1 | 20 | 11 | 14 |

Die Bieter teilen ihren Wert am Anfang mit durch ein "versiegeltes" Gebot. Der Mechanismus wählt den Teilnehmer mit höchstem Gebot als Gewinner. Anschließend muss eine Zahlung bestimmt werden. Wie ist die Zahlung zu wählen, damit die Bieter von vornherein die wahren Gebote abgeben möchten?

Angenommen, die Bieter müssen nichts zahlen. Dann versuchen sie unbeschränkt hohe Gebote abzugeben, da sie den Gegenstand umsonst erhalten können.

Alternativ können wir die Zahlungen mit den Geboten gleichsetzen. Der Bieter mit höchstem Wert wird versuchen, das zweithöchste Gebot zu erraten und ein wenig höher zu bieten. Wir können die Zahlung des Siegers auf das zweithöchste Gebot setzen. Dieser Ansatz nennt sich Vickrey (Zweitpreis-)Auktion.

|         |   |   |    |    |    |
|---------|---|---|----|----|----|
| Wert    | 9 | 1 | 20 | 11 | 14 |
| Zahlung | 0 | 0 | 14 | 0  | 0  |
| Nutzen  | 0 | 0 | 6  | 0  | 0  |

■

Ein Mechanismus heißt *anreizkompatibel*, wenn für jeden Bieter  $i$  und alle Gebote anderer Bieter eine ehrliche Offenlegung immer maximalen Nutzen für  $i$  liefert.

**Proposition 6.2.** *Die Vickrey-Auktion ist anreizkompatibel.*

Diese Eigenschaft wird an einem Beispiel deutlich:

|         |   |    |    |   |    |
|---------|---|----|----|---|----|
| Wert    | ? | ?  | 20 | ? | ?  |
| Gebot   | 5 | 11 | x  | 2 | 14 |
| Zahlung |   |    | 14 |   |    |
| Nutzen  |   |    | 6  |   |    |

Fall 1:  $i$  gewinnt mit ehrlichem Gebot  $x = 20$ , dann hat er für alle  $x \geq 14$  Nutzen 6; für  $x < 14$  Nutzen 0.

|         |   |    |    |   |    |
|---------|---|----|----|---|----|
| Wert    | ? | ?  | 20 | ? | ?  |
| Gebot   | 5 | 11 | x  | 2 | 24 |
| Zahlung |   |    | 0  |   |    |
| Nutzen  |   |    | 0  |   |    |

Fall 2:  $i$  verliert mit ehrlichem Gebot  $x = 20$ , dann hat er für alle  $x < 24$  Nutzen 0; für  $x \geq 24$  Nutzen  $-4$ .

**Definition 6.3** (Mechanismen mit direkter Offenlegung).

Wir verwenden folgende Notation:

- $V = V_1 \times \dots \times V_n$  und  $v \in V$
- $v = (v_1, \dots, v_n)$ ,  $v_i$  ist *Typ* von Bieter  $i$
- Bieter "bietet": Teilt einen Typ  $b_i \in V_i$  an den Mechanismus mit
- Ergebnisfunktion  $f : V \rightarrow A$ , Zahlungsfunktionen  $p_1, \dots, p_n$
- $p_i : V \rightarrow \mathbb{R}$  gibt die Zahlungen an, die Bieter  $i$  leisten muss.

**Definition 6.4** (anreizkompatibler (incentive compatible) Mechanismus).

- Betrachte jeden Bieter  $i$ , jede Bewertung  $v \in V$ , und jede alternative Bewertung  $v'_i \in V_i$ .
- Wir schreiben für die Ergebnisse  $x = f(v_i, b_{-i})$  und  $y = f(v'_i, b_{-i})$

- Mechanismus  $(f, p_1, \dots, p_n)$  ist anreizkompatibel, wenn der Nutzen

$$v_i(x) - p_i(v_i, b_{-i}) \geq v_i(y) - p_i(v'_i, b_{-i})$$

**Beispiel 6.5.** *Scaled-Bid Auktion*

| Bieter | 1 | 2 | 3  | 4  | 5  |
|--------|---|---|----|----|----|
| Wert   | 9 | 1 | 20 | 11 | 14 |

| Ergebnis | 1 | 2 | 3 | 4 | 5 |
|----------|---|---|---|---|---|
| $v_1$    | 9 | 0 | 0 | 0 | 0 |
| $v_2$    | 0 | 1 | 0 | 0 | 0 |
| etc.     |   |   |   |   |   |

Es gibt die Ergebnisse  $A = \{1, 2, 3, 4, 5\}$ , wobei  $i$  bedeutet “ $i$  gewinnt”. Die Ergebniswahl wird durch die Funktion ausgedrückt:  $f(b) = \arg \max_i \{b_i(i)\}$ . Die Zahlungen werden gesetzt als:  $p_i(b) = 0$  wenn  $f(b) \neq i$ , sonst  $p_i(b) = \max_{j \neq i} b_j(j)$ . ■

Wir können also Auktionen mit einem Gut anreizkompatibel gestalten. Wie kann man dieses Ergebnis verallgemeinern?

**Definition 6.6** (Vickrey-Clarke-Groves (VCG) Mechanismus). Ein *Vickrey-Clarke-Groves (VCG) Mechanismus* ist gegeben durch

- $f(b) \in \arg \max_{a \in A} \sum_i b_i(a)$ ; und
- es gilt für jedes  $b \in V$  und jeden Bieter  $i$

$$p_i(b) = h_i(b_{-i}) - \sum_{j \neq i} b_j(f(b)) ,$$

mit  $h_1, \dots, h_n$  beliebigen Funktionen  $h_i : V_{-i} \rightarrow \mathbb{R}$ .

**Beobachtung:** Der VCG-Mechanismus wählt ein Ergebnis  $a$ , das den *sozialen Nutzen*  $\sum_j b_j(a)$  maximiert. Die Funktion  $h_i$  hängt nicht vom eigenen “Gebot”  $v_i$  ab. Der Nutzen von Bieter  $i$  bei  $f(b) = x$  beträgt:

$$v_i(x) - p_i(b) = v_i(x) - h_i(b_{-i}) + \sum_{j \neq i} b_j(x) .$$

**Satz 6.7.** *Jeder VCG-Mechanismus ist anreizkompatibel.*

*Beweis.* Für die Bewertungen  $v$  sei  $v'_i \neq v_i$  eine “Lüge” für Bieter  $i$  und seien  $x = f(v_i, b_{-i})$  und  $y = f(v'_i, b_{-i})$  die beiden möglichen Ergebnisse.

Der Nutzen von  $i$ , wenn er  $v_i$  mitteilt, beträgt  $v_i(x) + \sum_{j \neq i} b_j(x) - h_i(b_{-i})$ . Wenn  $i$  hingegen lügt und  $v'_i$  mitteilt, dann erhält er den Nutzen  $v_i(y) + \sum_{j \neq i} b_j(y) - h_i(b_{-i})$ .

Der Nutzen von  $i$  wird also maximiert, wenn das Ergebnis den sozialen Nutzen  $v_i + \sum_{j \neq i} b_j$  maximiert.

Der VCG-Mechanismus wählt das Ergebnis, so dass der soziale Nutzen maximiert wird,  $\sum_i b_i(a)$ . Wenn  $b_i = v_i$ , dann ist  $\sum_i b_i(a) = v_i(a) + \sum_{j \neq i} b_j(a)$ , d.h. Mechanismus und Bieter  $i$  verfolgen das gleiche Ziel, und Ergebnis  $x$  ist die optimale Wahl für beide.

Angenommen Bieter  $i$  teilt  $v'_i$  mit, dann wählt VCG  $x$ .  $x$  gibt den optimalen sozialen Nutzen, wenn  $i$  lügt, aber evtl. einen suboptimalen echten Nutzen für  $i$ .

Der VCG-Mechanismus passt die Anreize der Bieter an den sozialen Nutzen an. □

Zusätzlich zur Anreizkompatibilität sollten wir noch weitere Eigenschaften fordern:

**Definition 6.8** (Individuell rational). Ein Mechanismus ist (ex-post) *individuell rational*, wenn die Bieter immer nicht-negativen Nutzen erhalten. Für alle  $b \in V$  und jeden Bieter  $i$  gilt  $v_i(f(b)) - p_i(b) \geq 0$ .

**Definition 6.9** (Positiver Transfer). Ein Mechanismus macht *keine positiven Transfers*, wenn kein Bieter jemals Geld ausgezahlt bekommt. Für alle  $b \in V$  und jeden Bieter  $i$  gilt  $p_i(b) \geq 0$ .

Wie muss die Funktion  $h$  gewählt werden, um diese Eigenschaften zu erfüllen?

**Definition 6.10** (Clarke-Regel). Die Funktion  $h_i(b_{-i}) = \max_{a \in A} \sum_{j \neq i} v_j(a)$  heißt Clarke-Pivot-Regel.

Mit der Clarke-Regel ergeben sich die Zahlungen von Bieter  $i$  als

$$p_i(b) = \max_{a \in A} \sum_{j \neq i} b_j(a) - \sum_{j \neq i} b_j(f(b)) .$$

Die Zahlung ist der "Gesamtschaden", den alle anderen Bieter in ihren genannten Bewertungen des Ergebnisses erfahren durch die Anwesenheit von  $i$ . Jeder Bieter *internalisiert die Externalitäten*.

**Lemma 6.11.** *Ein VCG-Mechanismus mit Clarke-Regel macht keine positiven Transfers. Wenn  $v_i(a) \geq 0$  für alle  $v_i \in V_i$  und  $a \in A$ , dann ist der Mechanismus individuell rational.*

*Beweis.* Seien  $x = f(b)$  und  $y = \arg \max_{a' \in A} \sum_{j \neq i} b_j(a')$  die beiden möglichen Ergebnisse. Per Definition gibt es keine positiven Transfers:

$$\sum_{j \neq i} b_j(y) - \sum_{j \neq i} b_j(x) \geq 0 .$$

Wenn die Bewertungen positiv sind, ist der Mechanismus individuell rational, da mit ehrlichem Gebot  $b_i = v_i$ :

$$v_i(x) + \sum_{j \neq i} b_j(x) - \sum_{j \neq i} b_j(y) \geq \sum_j b_j(x) - \sum_j b_j(y) \geq 0 .$$

□

**Beispiel 6.12.** *Bilateraler Handel*

|           | Handel | kein Handel |
|-----------|--------|-------------|
| Verkäufer | $-v_s$ | 0           |
| Käufer    | $v_b$  | 0           |

In diesem Beispiel gibt es einen Händler, der seine Ware an einen Kunden verkaufen will. Beide besitzen eine Bewertung für das gegebene Gut.

Bei welchen Bewertungen des Gutes sollte der Handel stattfinden?

Der Handel findet statt, wenn  $v_b > v_s$ , kein Handel, wenn  $v_s > v_b$  (für  $v_s = v_b$  wird ein Tie-Break bestimmt).

Wir analysieren den VCG-Mechanismus – er sollte den Handel nicht subventionieren.

- VCG Zahlungen bei ehrlichen Geboten für keinen Handel:  
Zahlungen Verkäufer:  $h_s(v_b) - 0$ , Zahlungen Käufer:  $h_b(v_s) - 0$   
Keine zusätzlichen Zahlungen durch den Mechanismus, daher  $h_s(v_b) = h_b(v_s) = 0$ .
- VCG Zahlungen bei ehrlichen Geboten für Handel:  
Zahlungen Verkäufer:  $h_s(v_b) - v_b$ , Zahlungen Käufer:  $h_b(v_s) + v_s$   
Verkäufer bekommt  $v_b$ , aber Käufer zahlt nur  $v_s < v_b$ .

Das *Budget ist unbalanciert*: Der VCG-Mechanismus subventioniert den Handel! ■

### 6.1.1 Rückwärtsauktion

Bei einer Rückwärtsauktion kauft der Auktionator einen Service. Jeder Teilnehmer bietet den Service an und hat interne Kosten. Der Auktionator bezahlt die Teilnehmer, es gibt also einen negativen Nutzen und negative Zahlungen.

Bei einer Vickrey-Rückwärts-Auktion gewinnt der Teilnehmer mit niedrigstem Gebot. Die Zahlung an den Gewinner ist das zweitniedrigste Gebot.

**Korollar 6.13.** Die Vickrey-Rückwärts-Auktion ist anreizkompatibel.

Dies wird durch das folgende Beispiel verdeutlicht:

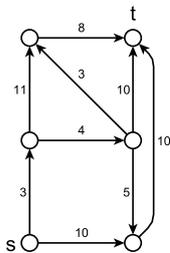
- Fall 1: Mit ehrlichem Gebot gewinnt Bieter  $i$ .

|         |    |     |    |     |     |
|---------|----|-----|----|-----|-----|
| Wert    | ?  | ?   | -7 | ?   | ?   |
| Gebot   | -9 | -11 | x  | -17 | -14 |
| Zahlung |    |     | -9 |     |     |
| Nutzen  |    |     | 2  |     |     |

Fall 2: Mit ehrlichem Gebot verliert Bieter  $i$ .

|         |    |     |     |     |     |
|---------|----|-----|-----|-----|-----|
| Wert    | ?  | ?   | -12 | ?   | ?   |
| Gebot   | -9 | -11 | x   | -17 | -24 |
| Zahlung |    |     | 0   |     |     |
| Nutzen  |    |     | 0   |     |     |

**Beispiel 6.14.** Kaufen eines Pfades im Netzwerk, Rückwärts-Auktion



Teilnehmer sind die Kanten in einem Netzwerk. Jede Kante  $e$  hat einen privaten Kostenwert  $c_e$ . Der Auktionator kauft einen  $s-t$ -Pfad.

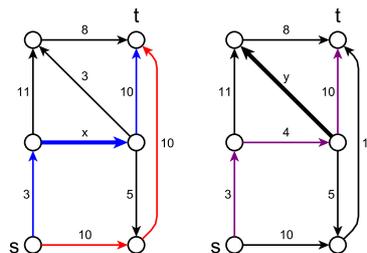
Die möglichen Ergebnisse sind die  $s-t$ -Pfade im Graph  $G$ . Der VCG-Mechanismus wählt den günstigsten Pfad  $P^*$  bzgl. der mitgeteilten Kantenkosten  $c_e$ , also den besten Pfad bzgl. der Werte  $v_e = -c_e$ . Die Zahlung der Kante  $e \in P^*$  an den Mechanismus lautet:

$$h_e(c_{-e}) - \sum_{e' \in P^*, e' \neq e} -c_{e'} = h_e(c_{-e}) + c(P^* - e)$$

Wir nutzen hier  $h_e(c_{-e}) = \max_{P \in G - e} \sum_{e \in P} -c_e = -c(P_{-e}^*)$ , wobei  $P_{-e}^*$  ein kürzester  $s-t$ -Pfad, wenn  $G$  die Kante  $e$  nicht enthalten würde. Diese Wahl für  $h_e$  ist nicht exakt die Clarke-Regel, da nicht alle Pfade disjunkt sind.

Die gesamte Zahlung der Kante  $e \in P^*$  ist also  $c(P^* - e) - c(P_{-e}^*) \leq 0$ , d.h. die Kante bekommt eine Zahlung vom Mechanismus. Jede Kante  $e \notin P^*$  hat Kosten 0 und erhält keine Zahlungen.

Wieder ist "Sag-die-Wahrheit" eine dominante Strategie.



Der VCG-Mechanismus ist anreizkompatibel und maximiert den sozialen Nutzen  $f$ . Welche anderen Ergebnisfunktionen  $f$  können wir *implementieren*, d.h. durch Zahlungen in anreizkompatible Mechanismen verwandeln?

Gibt es noch andere Arten von anreizkompatiblen Mechanismen außer VCG?

## 6.2 Direkte Charakterisierung

**Proposition 6.15.** *Ein Mechanismus ist anreizkompatibel genau dann, wenn für jeden Bieter  $i$  und jedes  $b_{-i}$  gilt:*

1. Die Zahlung  $p_i$  hängt nicht von  $b_i$  ab, sondern nur vom Ergebnis – d.h. es gibt Preise  $p_a(b_{-i}) \in \mathbb{R}$ , so dass für jedes  $b_i$  mit  $f(b_i, b_{-i}) = a$  gilt  $p_i(b_i, b_{-i}) = p_a(b_{-i})$ .
2. Der Mechanismus optimiert den Nutzen für jeden Bieter – d.h. für jedes  $b_i$  gilt  $f(b_i, b_{-i}) \in \arg \max_{a \in A'} \{b_i(a) - p_a\}$ , wobei  $A'$  die Menge der möglichen Ergebnisse für  $f(\cdot, b_{-i})$ .

*Beweis.* Offensichtlich: Bedingungen erfüllt  $\Rightarrow$  Anreizkompatibel.

1. Zahlung  $p_i = p_a$  hängt nicht ab von  $b_i$ , nur vom Ergebnis  $a = f(b_i, b_{-i})$ .
2. Mechanismus optimiert für jeden Bieter.

Anreizkompatibel  $\Rightarrow$  Bedingungen erfüllt:

- Bedingung 1:  
Angenommen  $b_i \neq b'_i$  ergeben das gleiche Ergebnis bei festem  $b_{-i}$ . Zahlung  $p_i(b_i, b_{-i}) > p_i(b'_i, b_{-i})$ , dann will Bieter  $i$  mit  $v_i = b_i$  lieber lügen und  $b'_i$  mitteilen.
- Bedingung 2:  
Falls der Mechanismus nicht den Nutzen für jeden Spieler maximiert, dann gibt es ein besseres Ergebnis  $a' \in \arg \max_a (b_i(a) - p_a)$  und ein  $b'_i$  mit  $a' = f(b'_i, b_{-i})$ . Also will Bieter  $i$  mit  $b_i = v_i$  lieber lügen und  $b'_i$  mitteilen.

□

**Definition 6.16** (Affiner Maximierer). Eine Ergebnisfunktion  $f$  ist ein *affiner Maximierer*, wenn es eine Teilmenge  $A' \subseteq A$ , Bietergewichte  $w_1, \dots, w_n \geq 0$  und Ergebnisgewichte  $c_a \in \mathbb{R}$  für jedes  $a \in A'$  gibt, so dass

$$f(b_1, \dots, b_n) \in \arg \max_{a \in A'} \left\{ c_a + \sum_i w_i b_i(a) \right\}.$$

**Proposition 6.17.** *Sei  $f$  ein affiner Maximierer, und sei  $h_i$  eine beliebige Funktion unabhängig von  $b_i$ . Bieter  $i$  mit  $w_i = 0$  zahlt  $p_i(b) = 0$ . Bieter  $i$  mit  $w_i > 0$  zahlt*

$$p_i(b) = h_i(b_{-i}) - \frac{1}{w_i} \left( \sum_{j \neq i} w_j b_j(a) + c_a \right).$$

Dann ist  $(f, p_1, \dots, p_n)$  anreizkompatibel.

*Beweis.* Für  $w_i = 0$  hat  $i$  keinen Einfluß auf den Mechanismus. Mit  $p_i = 0$  ergibt sich der gleiche Nutzen für jedes Gebot von  $i$ .

Für  $w_i > 0$  sei o.B.d.A.  $h_i = 0$ . Der Nutzen von  $i$ , wenn  $a$  gewählt wird, beträgt:

$$v_i(a) + \frac{1}{w_i} \left( \sum_{j \neq i} w_j b_j(a) + c_a \right).$$

Multipliziere mit  $w_i > 0$ , dann ist der Ausdruck maximal, wenn  $c_a + \sum_i w_i v_i(a)$  maximal ist.  $f$  ist ein affiner Maximierer,  $b_i = v_i$  optimiert den Nutzen für Mechanismus und Bieter  $i$ . Der wahre Typ ist eine dominante Strategie für  $i$ .  $\square$

**Satz 6.18** (Roberts 1979). *Seien  $|A| \geq 3$ ,  $f$  bildet voll auf  $A$  ab,  $V_i = \mathbb{R}^A$  für jeden Bieter  $i$ , und  $(f, p_1, \dots, p_n)$  anreizkompatibel. Dann ist  $f$  ein affiner Maximierer.*

## 6.3 Single-Parameter Mechanismen

Im Single-Parameter-Fall haben die Bewertungen eine einfache Struktur.

Im Ergebnis  $a \in A$  erhält Bieter  $i$  eine Menge von "Zeugs". Sei  $x_i(a) \in \mathbb{R}$  die Menge an "Zeugs", die Bieter  $i$  in Ergebnis  $a$  erhält. Die Menge an "Zeugs" wird mit einem einzelnen Parameter bewertet: Der private Wert pro Einheit Zeugs beträgt  $t_i \in \mathbb{R}$ , die Bewertungsfunktion ist definiert als  $v_i(a) = t_i \cdot x_i(a)$ .

**Definition 6.19** (Single-Parameter Bereich). Ein *Single-Parameter Bereich*  $V_i$  ist gegeben durch die (öffentlich bekannte) Funktion  $x_i : A \rightarrow \mathbb{R}$  und den Wertebereich  $[t_i^0, t_i^1]$ . Die Menge  $V_i$  enthält alle  $v_i$ , so dass für ein  $t_i^0 \leq t_i \leq t_i^1$  gilt

$$v_i(a) = t_i \cdot x_i(a) .$$

Wir überladen die Notation:  $v_i$  bezeichnet sowohl Funktion als auch Parameter.

Im Single-Parameter-Bereich gibt es einfache Beispiele:

- Ein-Gut-Auktion:  $x_i(a) \in \{0, 1\}$  und  $\sum_i x_i(a) \leq 1$ .
- $k$  identische Güter:  $x_i(a) \in \{0, 1, \dots, k\}$  und  $\sum_i x_i(a) \leq k$ .
- $s$ - $t$ -Pfad:  $x_e(a) \in \{0, 1\}$  und  $P(x) = \{e \mid x_e(a) = 1\}$  ist  $s$ - $t$ -Pfad in  $G$ .

**Beispiel 6.20.** *Sponsored-Search-Auktion*

Eine Webseite mit Suchresultaten hat mehrere Slots für Werbeanzeigen. Die Suchmaschine versteigert diese Anzeigeslots an Werbekunden, wobei Slot  $k$  eine bekannte *Anklickrate*  $\alpha_k \geq 0$  hat.

Jede Firma  $i$  hat den privaten Wert  $v_i$  pro Klick auf ihre Anzeige. Im Ergebnis  $a \in A$  werden Anzeigeslots an Firmen zugewiesen. Dabei gilt  $x_i(a) = \alpha_k$ , wenn Firma  $i$  einen Slot  $k$  erhält, sonst  $x_i(a) = 0$ . Die Firmen bewerten das Ergebnis  $a$  als  $v_i(a) = v_i \cdot x_i(a)$ .  $\blacksquare$

Gibt es anreizkompatible Mechanismen, die keine affinen Maximierer sind?

Wir versteigern ein einzelnes Gut und geben es dem *zweithöchsten* Bieter. Gibt es Zahlungen, mit denen der Mechanismus anreizkompatibel wird?

Betrachte einen Bieter  $i$  und fixiere die anderen Gebote  $b_{-i}$ .

Es gilt  $x_i(a) \in \{0, 1\}$ . Die direkte Charakterisierung zeigt:  $i$  zahlt immer nur  $p_i^1$  oder  $p_i^0$ , je nachdem ob er zweithöchster Bieter ist oder nicht.

Sei  $y$  ein Gebot, mit dem  $i$  zweithöchster Bieter wird, und  $z$  eines mit dem er höchster Bieter wird, mit  $y < z$ .

Wenn  $v_i = y$ , soll  $i$  nicht  $z$  lügen, also:  $y \cdot 1 - p_i^1 \geq y \cdot 0 - p_i^0$ .

Wenn  $v_i = z$ , soll  $i$  nicht  $y$  lügen, also:  $z \cdot 0 - p_i^0 \geq z \cdot 1 - p_i^1$ .

Daraus folgt  $y \geq z$ , ein Widerspruch.

Es gibt keine Zahlungen, so dass der Mechanismus anreizkompatibel wird. Die Ergebnisfunktion ist *nicht monoton* – ein höheres Gebot kann die Menge an erhaltenem Zeugs verringern.

**Definition 6.21** (Monotonie). Eine Ergebnisfunktion  $f$  für Single-Parameter Bereiche  $V_1, \dots, V_n$  ist *monoton in  $b_i$* , wenn für jedes  $b_{-i}$  und jedes  $b'_i \in V_i$  mit  $b'_i \geq b_i$  gilt

$$x_i(f(b'_i, b_{-i})) \geq x_i(f(b_i, b_{-i})) .$$

Ein Mechanismus ist *normalisiert*, wenn beim kleinsten Gebot  $t_i^0$  Bieter  $i$  nie Zeugs bekommt und nichts zahlt, d.h.  $x_i(t_i^0, b_{-i}) = 0$  und  $p_i(t_i^0, b_{-i}) = 0$  für alle  $b_{-i}$ .

**Satz 6.22** (Myersons Lemma). *Ein normalisierter Mechanismus  $(f, p_1, \dots, p_n)$  für Single-Parameter Bereiche ist anreizkompatibel genau dann, wenn die folgenden Bedingungen gelten:*

- Die Ergebnisfunktion  $f$  ist monoton in jedem  $b_i$ .
- Die Zahlungen sind gegeben durch

$$p_i(b_i, b_{-i}) = b_i \cdot x_i(f(b)) - \int_{t_i^0}^{b_i} x_i(f(t, b_{-i})) dt .$$

*Beweis.* Fixiere  $b_{-i}$ . Seien  $y < z$  zwei mögliche private Werte von  $i$ . Wir schreiben  $a_y = f(y, b_{-i})$  und  $a_z = f(z, b_{-i})$ . Anreizkompatibel bedeutet:

$$y \cdot x_i(a_y) - p_i(a_y) \geq y \cdot x_i(a_z) - p_i(a_z) \quad (6.1)$$

und

$$z \cdot x_i(a_z) - p_i(a_z) \geq z \cdot x_i(a_y) - p_i(a_y) \quad (6.2)$$

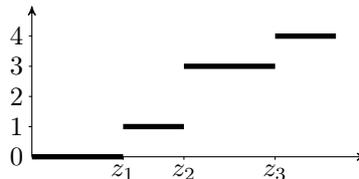
Summiere (6.1) und (6.2) und stelle um:

$$z \cdot (x_i(a_z) - x_i(a_y)) \geq y \cdot (x_i(a_z) - x_i(a_y))$$

Da  $z > y$ , folgt  $x_i(a_z) \geq x_i(a_y)$ , also  $f$  monoton ist notwendig.

Ist  $f$  monoton auch hinreichend? Wenn  $f$  monoton, dann müssen die Zahlungen zumindest genau wie in der Formel sein – wir beweisen das nur im Spezialfall für  $x_i(a) \in \mathbb{N}$ .

Sei  $x_i$  monoton und  $x_i(a) \in \{0, 1, 2, \dots, k\}$ , eine Treppenfunktion.  $x_i$  springt bei  $z_1 \leq z_2 \leq \dots \leq z_\ell$  um  $k_1, k_2, \dots, k_\ell$ , wobei  $\sum_{j=1}^{\ell} k_j \leq k$ .

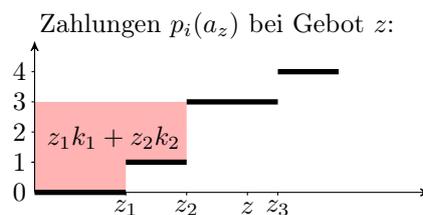
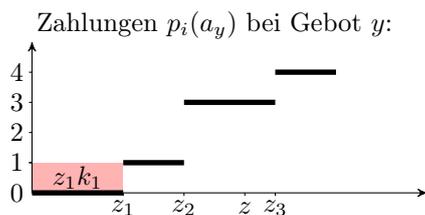


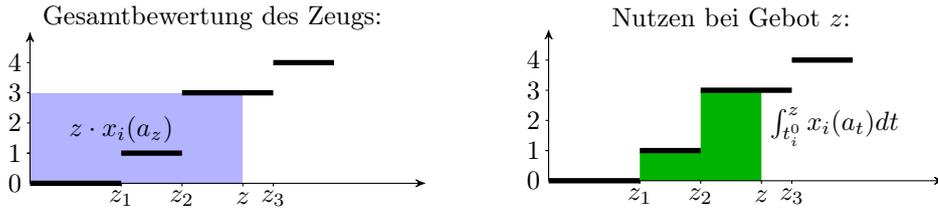
Die Ungleichungen (6.1) und (6.2) liefern

$$z \cdot (x_i(a_z) - x_i(a_y)) \geq p_i(a_z) - p_i(a_y) \geq y \cdot (x_i(a_z) - x_i(a_y)) .$$

Also ist  $p_i(a_z) = p_i(a_y)$ , wenn  $x_i(a_z) = x_i(a_y)$ . Sei  $z = z_i$  und  $y = z_i - \varepsilon$ , dann zeigt  $\varepsilon \rightarrow 0$ , dass  $p_i$  bei  $z_i$  um  $z_i k_i$  springt. Damit

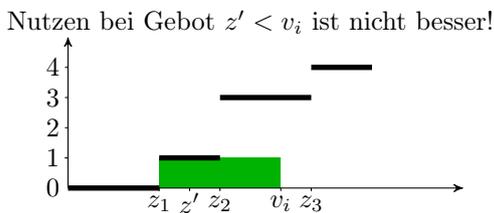
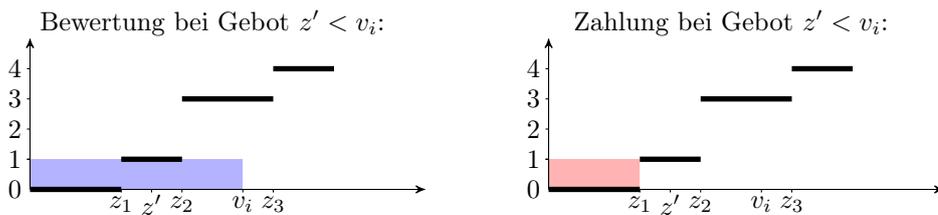
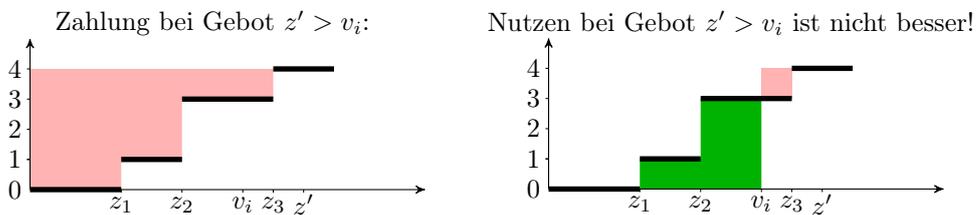
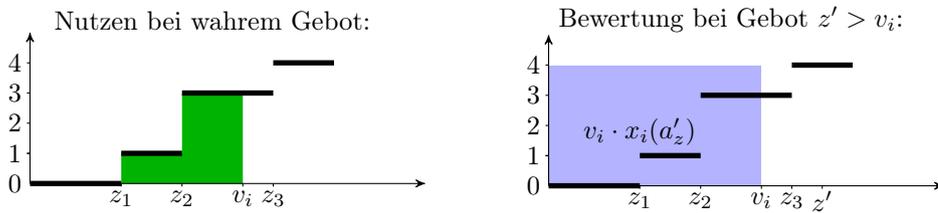
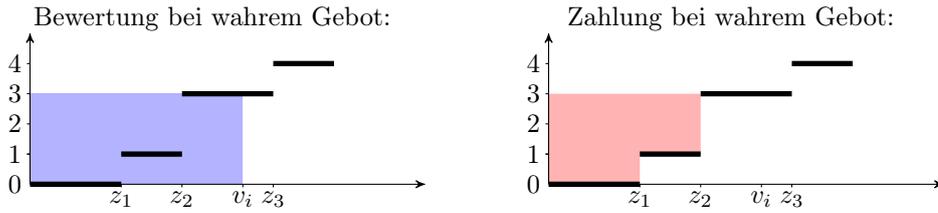
$$p_i(a_z) = \sum_{j: z_j \leq z} z_j k_j = z \cdot x_i(a_z) - \int_{t_i^0}^z x_i(a_t) dt .$$





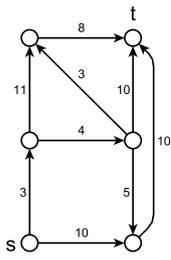
Die Zahlungen für die monotone Funktion  $f$  lassen sich mit Hilfe der Gesamtbewertung und des Nutzens formulieren.

Ist der Mechanismus nun wirklich immer anreizkompatibel, d.h. ist monotonen  $f$  hinreichend?



□

**Beispiel 6.23.** Rückwärtsauktion und Min-Max-Pfade



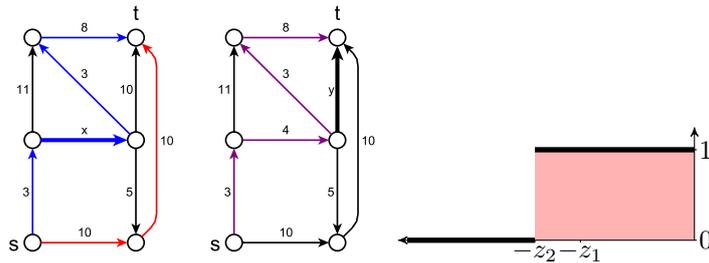
Die Teilnehmer entsprechen den Kanten in einem Netzwerk, dabei besitzt jede Kante  $e$  einen privaten Kostenwert  $c_e$ . Der Auktionator kauft einen  $s$ - $t$ -Pfad.

Der gekaufte Pfad  $P^*$  soll den *maximalen Kostenwert einer Kante minimieren*.

Wir suchen einen anreizkompatiblen Mechanismus mit einer monotonen Kostenfunktion.

Verringert  $e$  sein Gebot, kann er nur in  $P^*$  kommen oder bleiben. Da  $x_i(f(v_i, v_{-i})) \in \{0, 1\}$ , gibt es höchstens einen Treppenschritt.

Mit Myersons Lemma ergeben sich die Zahlungen für den anreizkompatiblen Mechanismus:  $e \notin P^*$  erhält keine Zahlung.  $e \in P^*$  erhält die maximalen Kantenkosten auf dem Min-Max-Pfad in  $G - \{e\}$  als Zahlungen.



Durch die negativen Zahlungen dreht sich unsere Perspektive. Weiterhin bezahlen die Teilnehmer den Wert, bei dem die Ergebnisfunktion von 0 auf 1 springt. ■

### 6.4 Revelationsprinzip

Alle bisherigen Resultate betreffen Mechanismen mit direkter Offenlegung. Kann man mit komplizierterer Kommunikation noch grundsätzlich andere Mechanismen entwerfen? Zum Beispiel könnte ein Mechanismus in  $k$  Runden nacheinander jedem Bieter bestimmte Ja/Nein-Fragen stellen, und die Bieter müssten darauf reagieren. Oder ein Mechanismus würde in jeder Runde zwei Ergebnisse präsentieren und jeden Bieter fragen, welches Ergebnis er besser findet. Oder ein anderes Interaktionsprotokoll wird umgesetzt, oder...

Für allgemeine Kommunikation zwischen Mechanismus und Bieter  $i$  nehmen wir an, es gibt für jeden Bieter  $i$  eine Menge  $B_i$  von *möglichen Aktionen*, wobei jedes  $b_i \in B_i$  eine *Kollektion von Antworten* darstellt, die Bieter  $i$  auf die Fragen des Mechanismus übermitteln kann.

**Definition 6.24** (Allgemeiner Mechanismus mit Aktionsraum). Sei  $B_i$  der *Aktionsraum* für Bieter  $i$ , so dass  $B = B_1 \times \dots \times B_n$ . Eine *Strategie*  $s_i : B_i \rightarrow X_i$  bildet die private Bewertung des Bieters  $v_i \in V_i$  ab auf eine Aktion. Jeder Bieter  $i$  wählt eine Strategie  $s_i$  und damit eine Aktion  $b_i = s_i(v_i)$ . Die *Ergebnisfunktion*  $g : X \rightarrow A$  bildet die gewählten Aktionen auf ein Ergebnis ab. Abhängig von den gewählten Aktionen ergeben sich die *Zahlungen*  $p_i : B \rightarrow \mathbb{R}$  und der quasi-lineare *Nutzen*:  $u_i(x) = v_i(g(x)) - p_i(x)$ .

Bei direkter Offenlegung gilt  $B_i = V_i$ . Hier teilt ein Bieter mit seiner Strategie direkt seine (evtl. gelogene) Bewertung mit. Allgemein sind die Antworten  $B_i$  aber nicht unbedingt identisch mit den Bewertungen  $V_i$ . Mit der Strategie legt der Bieter für jede mögliche Bewertung fest, welche Antworten er dem Mechanismus gibt.

Sei nun das Strategieprofil  $s(v) = (s_1, \dots, s_n)$  ein Gleichgewicht  $s$  in dominanten Strategien im allgemeinen Mechanismus. Sei  $f(v) = g(s(v))$ . Wir sagen der Mechanismus *implementiert die Ergebnisfunktion  $f$  in dominanten Strategien*.

Bei anreizkompatiblen Mechanismen mit direkter Offenlegung ist Sag-die-Wahrheit für jeden Bieter eine dominante Strategie. Formal gibt es in solchen Mechanismen also ein Gleichgewicht  $s$  in dominanten Strategien mit  $b_i = s_i(v_i) = v_i$  für alle  $v_i \in V_i$  und alle Bieter  $i$ .

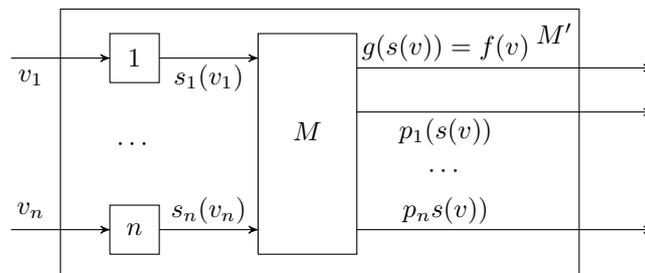
Das Revelationsprinzip sagt, dass man durch komplizierte Kommunikation keine grundsätzlich anderen Mechanismen mit dominanten Strategien erzeugen kann. Wir können uns also weiterhin auf Mechanismen mit direkter Offenlegung konzentrieren.

**Proposition 6.25** (Revelationsprinzip). *Es gibt einen allgemeinen Mechanismus  $M$ , der  $f$  in dominanten Strategien implementiert.*

$\Leftrightarrow$

*Es gibt einen anreizkompatiblen Mechanismus  $M'$  mit direkter Offenlegung und Ergebnisfunktion  $f$ .*

*Beweis.* Der allgemeine Mechanismus  $M$  verwendet ein Kommunikations-Protokoll für die Bestimmung des Ergebnisses. Die Bieter wählen ihre wahren Strategien  $s_i(v_i)$  und teilen diese  $M$  mit. Wir entwerfen einen anreizkompatiblen Mechanismus  $M'$  mit direkter Offenlegung, dem die Spieler nur ihr wahres Gebot mitteilen. Der Mechanismus simuliert mit Hilfe der Programme  $i$  intern das Protokoll von  $M$ , wobei die Programme  $i$  die Strategien  $s_i(v_i)$  für die Bieter wählen.



□

## 6.5 Komplexität

Aus Myersons Lemma folgt, dass sich der *Entwurf von anreizkompatiblen Mechanismen* auf den *Entwurf von monotonen Ergebnisfunktionen* reduziert. Dabei treten aber Probleme mit der Komplexität auf.

Wir betrachten ein erstes Beispiel.

### 6.5.1 Rucksackauktion

Ein TV-Sender möchte eine Werbepause von  $G$  Sekunden mit Spots füllen. Es gibt eine Menge  $I$  von  $n$  Firmen, die einen Spot senden wollen. Jede Firma  $i \in I$

- liefert einen Werbespot mit Länge  $g_i \leq G$  Sekunden ( $g_i$  öffentlich bekannt).
- hat eine Bewertung  $v_i \geq 0$ , wenn ihr Spot gesendet wird ( $v_i$  private Information), und Bewertung 0 sonst.

Die Rucksackauktion ist offensichtlich ein Single-Parameter Bereich. Betrachten wir zuerst den VCG-Mechanismus:

- Frage Bewertungen  $v_i$  von allen Firmen  $i \in I$  ab, Firma  $i$  teilt Gebot  $b_i$  mit.

- Wähle Teilmenge  $S \subseteq I$  von Spots, die den sozialen Nutzen maximiert:

$$f(v) = \arg \max_{S \subseteq I} \left\{ \sum_{i \in S} b_i \mid \sum_{i \in S} g_i \leq G \right\} .$$

- Wähle Zahlungen  $p_i(b)$  wie in Myersons Lemma.

VCG muss *optimale Lösungen für Instanzen des Rucksackproblems berechnen*. Aber das *Rucksackproblem ist NP-hart*. Damit ergibt sich ein Zielkonflikt zwischen drei wünschenswerten Eigenschaften des Mechanismus:

- (1) anreizkompatibel,
- (2) maximiert sozialen Nutzen,
- (3) berechenbar in polynomieller Zeit.

Der Konflikt besteht zwischen Eigenschaften (2) und (3). Die Problematik wird im Bereich der *Approximationsalgorithmen* seit Jahrzehnten erforscht. Wenn wir diese Algorithmen benutzen, schwächen wir Eigenschaft (2) ab zu “approximiert sozialen Nutzen möglichst gut”.

Wir können aber *nicht irgendwelche* Approximationsalgorithmen anwenden. Da wir (1) einhalten wollen, *müssen auch Zahlungen existieren*, die einen anreizkompatiblen Mechanismus ergeben. Im Single-Parameter Bereich müssen wir also *möglichst gute monotone* Approximationsalgorithmen entwerfen.

Eine zentrale Frage im Algorithmischen Mechanismusedesign lautet: Wieviel muss man durch die zusätzliche Forderung der Anreizkompatibilität an sozialem Nutzen verlieren?

Wie gut sind anreizkompatible Approximationsalgorithmen im Vergleich zu beliebigen Approximationsalgorithmen?

Wir bezeichnen eine optimale Teilmenge von Spots mit  $S^*$ .

**Definition 6.26** (*c*-Approximationsalgorithmus). Ein *c*-Approximationsalgorithmus liefert eine Teilmenge  $T \subseteq I$  mit

$$\sum_{i \in T} b_i \geq \frac{1}{c} \cdot \sum_{i \in S^*} b_i .$$

Eine einfache *n*-Approximation:

Wähle nur einen einzigen Spot mit maximaler Bewertung. Anreizkompatibel ist trivial – wir behandeln die gesamte Werbepause wie ein einzelnes Gut (also eine Vickrey-Zweitpreisauktion).

Das geht doch besser! In “Theoretische Informatik 1” haben wir dereinst bewiesen:

**Satz 6.27.** *Das Rucksackproblem hat ein volles Approximationsschema, d.h. für jedes  $\varepsilon > 0$  können wir eine  $(1 + \varepsilon)$ -approximative Lösung berechnen in Zeit  $O(n^3/\varepsilon)$ .*

Leider kann man für diesen Algorithmus zeigen: Er ist *nicht monoton*, siehe unten.

**Beispiel 6.28.** *Rucksackauktion einer Werbepause*

Sei die Gesamtlänge der Werbepause  $G = 100$  Sekunden.

| Firma       | 1  | 2  | 3  | 4  | 5  |
|-------------|----|----|----|----|----|
| $v_i = b_i$ | 45 | 20 | 45 | 40 | 50 |
| $g_i$       | 15 | 25 | 60 | 50 | 90 |

**Algorithmus 7:** Greedy-Algorithmus für die Rucksackauktion

---

**Input:**  $(g_i, v_i)$  für jede Firma  $i \in I$   
**Output:** Menge  $S$  von gewählten Spots

1 **Sortiere Firmen:**

$$\frac{b_1}{g_1} \geq \dots \geq \frac{b_n}{g_n}$$

2 **Setze**  $S' \leftarrow \emptyset$  **und**  $j \leftarrow 1$ , **sei**  $b_{\max} = \max_j b_j$   
3 **while**  $(g_j + \sum_{k \in S'} g_k) \leq G$  **do**  
4    $S' \leftarrow S' \cup \{j\}$  **und**  $j \leftarrow j + 1$   
5 **if**  $b_{\max} > \sum_{k \in S'} b_k$  **then**  
6    $S \leftarrow \arg \max_j b_j$   
7 **else**  
8    $S \leftarrow S'$

---

Nach der Sortierung in Schritt 1 ergibt sich die Reihenfolge der Firmen als (1,4,2,3,5):

$$45/15 \geq 40/50 = 20/25 \geq 45/60 \geq 50/90.$$

In der Schleife in Schritten 2-4 ergibt sich  $S' = \{1, 4, 2\}$ .

In Schritt 5 gilt

$$50 = v_{\max} < \sum_{j \in S'} v_j = 105.$$

Also ist das Ergebnis  $S = \{1, 4, 2\}$  mit Wert 105.

Optimum:  $S^* = \{1, 2, 3\}$  mit Wert 110.

Wir modifizieren die Werte, um Schritt 5 zu verdeutlichen:

|             |    |    |    |    |     |
|-------------|----|----|----|----|-----|
| Firma       | 1  | 2  | 3  | 4  | 5   |
| $v_i = b_i$ | 45 | 20 | 45 | 40 | 260 |
| $g_i$       | 15 | 25 | 60 | 50 | 90  |

Nach der Sortierung in Schritt 1 ergibt sich die Reihenfolge der Firmen als (1,5,4,2,3):

$$45/15 \geq 260/90 \geq 40/50 = 20/25 \geq 45/60.$$

In der Schleife in Schritten 2-4 ergibt sich  $S' = \{1\}$ .

In Schritt 5 gilt

$$260 = v_{\max} > \sum_{j \in S'} v_j = 45.$$

Also ist das Ergebnis  $S = \{5\}$  mit Wert 260.

Optimum:  $S^* = \{5\}$  mit Wert 260. ■

**Satz 6.29.** *Der Greedy-Algorithmus ist 2-approximativ und monoton. Es gibt für die Rucksackauktion einen anreizkompatiblen Mechanismus, der mindestens die Hälfte des optimalen sozialen Nutzens garantiert.*

*Beweis.* Man kann direkt beobachten, dass der Greedy-Algorithmus monoton ist (Übung).

Zur Beschränkung des Approximationsfaktors nutzen wir die fraktionale Relaxierung, bei der wir

von jedem Spot  $i$  einen beliebigen Bruchteil  $x_i \in [0, 1]$  senden können. Für die fraktionale Relaxierung optimieren wir also:

$$f_{\text{frak}}(v) = \arg \max_{x \in [0,1]^n} \left\{ \sum_i x_i b_i \mid \sum_i x_i g_i \leq G \right\} .$$

Die fraktionale Relaxierung hat eine Obermenge an Lösungen, daher gilt für die optimale fraktionale Lösung  $x^*$ , dass sie nur besser sein kann:

$$\sum_{i \in S^*} b_i \leq \sum_{i \in I} x_i^* b_i .$$

Wir versuchen in  $x^*$  für jede Sekunde der Werbepause *möglichst viel Wert pro Sekunde* rauszuholen. Seien die Spots nummeriert nach ihrem Wert pro Sekunde  $b_1/g_1 \geq \dots \geq b_n/g_n$ . Wir wählen so viele Sekunden wie möglich von Spot 1, dann so viele wie möglich von Spot 2 usw. bis  $G$  Sekunden gewählt sind.

Genau das macht auch der Greedy-Algorithmus in Schritt 2-4! Wenn Greedy abbricht, kann die fraktionale Lösung allerdings noch einen Bruchteil des nächsten Spots  $j'$  in der Reihenfolge dazunehmen:

$$\sum_{i \in I} x_i^* b_i = \sum_{k \in S'} 1 \cdot b_k + x_{j'}^* b_{j'} .$$

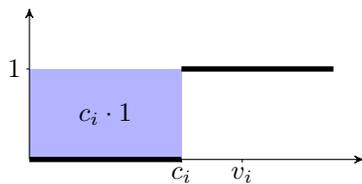
Also ergibt sich für den Approximationsfaktor

$$\begin{aligned} \frac{\sum_{k \in S^*} b_k}{\sum_{k \in S} b_k} &= \frac{\sum_{k \in S^*} b_k}{\max\{b_{\max}, \sum_{k \in S'} b_k\}} \leq \frac{\sum_{k \in S'} b_k + x_{j'}^* b_{j'}}{\max\{b_{\max}, \sum_{k \in S'} b_k\}} \\ &\leq 2 \cdot \frac{\sum_{k \in S'} b_k + x_{j'}^* b_{j'}}{\sum_{k \in S'} b_k + b_{\max}} \leq 2 . \end{aligned}$$

□

Wie viel sollten die Firmen bezahlen?

Jede Firma bekommt hier nur eine *binäre* Menge an Zeugs – im Ergebnis  $a \in A$  wird Spot  $i$  gesendet ( $x_i(a) = 1$ ) oder nicht ( $x_i(a) = 0$ ). Jeder anreizkompatible Mechanismus liefert eine *monotone, binäre Schrittfunktion*  $x_i$ . Der Wert, bei dem  $x_i$  von 0 auf 1 springt, heißt *kritischer Wert*  $c_i(b_{-i})$ .



Natürlich hängt er von den Geboten  $b_{-i}$  der anderen Firmen ab. Für einen normalisierten Mechanismus gilt  $p_i(b) = 0$ , wenn Spot  $i$  nicht gezeigt wird. Sonst ergibt Myersons Lemma  $p_i(b) = c_i(b_{-i}) \cdot 1$ , d.h.  $i$  zahlt (bei festen Geboten der anderen Firmen) den *Wert des kleinsten Gebots, das die Ausstrahlung des Spots garantiert*.

Betrachten wir nun nochmal das volle Approximationsschema für das Rucksackproblem.

---

#### Algorithmus 8: Volles Approximationsschema

---

**Input:**  $(g_i, b_i)$  für jede Firma  $i \in I$  und  $\varepsilon > 0$

**Output:** Menge  $S$  von gewählten Spots

Sei  $b_{\max} = \max_i b_i$  und  $s = \varepsilon \cdot b_{\max}/n$

Runde alle Bewertungen auf ganze Zahlen:  $b'_i = \lfloor b_i/s \rfloor$

Löse das Problem mit Werten  $b'_i$  optimal mit dyn. Prog.

Sei  $S'$  die optimale Lösung für gerundete Bewertungen

Setze  $S \leftarrow S'$ .

---

Die dynamische Programmierung in Schritt 3 läuft in Zeit  $O(n^2 \cdot \max_i b'_i)$ . Durch Runden gilt  $b'_i \in \{0, 1, \dots, \lfloor n/\varepsilon \rfloor\}$ , also läuft der Algorithmus für konstantes  $\varepsilon > 0$  in polynomieller Zeit  $O(n^3/\varepsilon)$ .

Das Schema ist nicht monoton, weil  $s$  vom höchsten Gebot  $b_{\max}$  abhängt. Wenn wir dagegen in Schritt 1 die Granularität auf eine Konstante  $s = \delta > 0$  unabhängig von  $b_1, \dots, b_n$  setzen, dann ist das Schema monoton (Übung).

Für welchen festen Wert  $\delta$  kann man ohne Wissen über die Bewertungen immer eine  $(1 + \varepsilon)$ -Approximation garantieren? Für gar keinen.

Deswegen rufen wir das Schema für *unendliche viele* konstante Werte für  $\delta$  auf. Dann wählen wir die *beste Lösung aus allen diesen Aufrufen*.

---

**Algorithmus 9:** Unendliches Schema

---

**Input:**  $(g_i, b_i)$  für jede Firma  $i \in I$  und  $\varepsilon > 0$

**Output:** Menge  $S$  von gewählten Spots

**for** alle  $k = \dots, -2, -1, 0, 1, 2, \dots$  **do**

Sei  $s(k) = \varepsilon \cdot 2^k/n$   
 Runde Bewertungen:  $b_i(k) = \min\{s(k) \cdot \lfloor b_i/s(k) \rfloor, 2^k\}$   
 Löse das Problem mit Werten  $b_i(k)$  optimal mit dyn. Prog.  
 Sei  $S(k)$  die optimale Lösung für gerundete Bewertungen

Setze  $S \leftarrow \arg \max_{S(k)} \sum_{i \in S(k)} b_i(k)$

(Tie-Breaking bzgl. kleinerem  $k$ )

---

Das Schema ist bei jedem Aufruf monoton in  $b_i$ . Der soziale Nutzen ist monoton in  $b_i$ . Deswegen ist auch die beste Lösung aus allen Aufrufen *monoton in  $b_i$* .

Sei  $k^* = \lceil \log_2(b_{\max}) \rceil$ , dann ist

$$\varepsilon \cdot b_{\max}/n \leq s(k^*) \leq \varepsilon \cdot 2 \cdot b_{\max}/n.$$

Also ist  $S'_{k^*}$  (und somit auch  $S$ ) höchstens eine  $(1 + 2\varepsilon)$ -Approximation.

Man kann zeigen, dass das unendliche Schema nur für relativ wenige Werte  $k \in \{k^* - \lceil \log_2 n \rceil - 2, \dots, k^*\}$  aufgerufen werden muss. Für andere Werte von  $k$  können sich keine besseren Lösungen ergeben. Wählen wir  $k$  größer  $k^*$ , wird die Granularität der Rundung vergrößert, so dass wir Informationen verlieren. Ist  $k$  kleiner als  $k^*$ , runden wir genauer, aber die gewonnenen Informationen werden mit kleinerem  $k$  immer unbedeutender.

Damit brauchen wird also *nicht unendliche viele Aufrufe* sondern nur *höchstens*  $\log_2(n) + 4$  *viele* für den passenden Wertebereich von  $k$ . Der passende Wertebereich hängt von  $k^*$  und damit von  $b_1, \dots, b_n$  ab. Das bedeutet aber nicht, dass wir  $k$  auf diesen Bereich einschränken – es besagt nur, dass die optimalen Lösungen über alle, unendlich viele konstante Werte von  $k$  in diesem Bereich gefunden werden. Damit sind die Monotonie-Argumente für feste Werte von  $k$  weiterhin gültig.

In jedem Aufruf benötigt die dynamische Programmierung eine Laufzeit  $O(n^2 \cdot \max_i b_i(k)/s(k))$ . Für den kleinsten betrachteten Wert  $k^* - \lceil \log_2 n \rceil - 2$  ergibt sich die kleinste Granularität und die größte Laufzeitschranke.

Es gilt

$$\begin{aligned} & \max_i \frac{b_i(k^* - \lceil \log_2 n \rceil - 2)}{s(k^* - \lceil \log_2 n \rceil - 2)} \\ & \leq \left\lfloor \frac{b_{\max}}{\varepsilon \cdot 2^{k^* - \lceil \log_2 n \rceil - 2}/n} \right\rfloor \leq \left\lfloor \frac{n \cdot 2^{k^*}}{\varepsilon \cdot 2^{k^* - \log_2(n) - 3}} \right\rfloor \\ & \leq \lfloor 8n^2/\varepsilon \rfloor. \end{aligned}$$

Damit hat die dynamische Programmierung in jedem der  $O(\log n)$  Aufrufe eine Laufzeit von höchstens  $O(n^4/\varepsilon)$ .

**Satz 6.30.** *Es gibt ein monotones volles Approximationsschema für das Rucksackproblem mit Laufzeit  $O(n^4 \log n/\varepsilon)$ . Für die Rucksackauktion gibt es anreizkompatible Mechanismen, die für jedes feste  $\varepsilon > 0$  mindestens einen  $1/(1 + \varepsilon)$ -Anteil des optimalen sozialen Nutzens garantieren.*

## 6.6 Ertragsmaximierung

Bisher war Geld nur *Mittel zum Zweck*, um den Mechanismus anreizkompatibel zu machen. Hier betrachten wir *Geld als Zielfunktion* des Mechanismus.

**Beispiel 6.31.** *Ein-Gut-Auktion mit einem Bieter*

Es gibt nur zwei mögliche Ergebnisse, und die Zahlung muss unabhängig vom Gebot des Bieters sein.

Anreizkompatible Mechanismen sind *Fixpreis-Mechanismen*:

- Wähle Preis  $p \geq 0$  (evtl. zufällig) *unabhängig vom Gebot*.
- Verkaufe das Gut, wenn Gebot  $b_i \geq p$ .

Der sozial Nutzen wird maximiert, wenn  $p = 0$ . Wie maximiert man den Ertrag? ■

Für Ertragsmaximierung brauchen wir (zumindest teilweise) Informationen über die möglichen Bewertungen der Bieter. Ansonsten kann der erzielte Ertrag beliebig niedriger sein als der optimale Ertrag.

Wir betrachten einen Single-Parameter Bereich für jeden Bieter  $i$ . Die *Verteilung*  $\mathcal{V}_i$  enthält alle privaten Parameter  $v_i \sim \mathcal{V}_i$  von Bieter  $i$ , wobei diese durch den *Verteilungsvektor*  $\mathcal{V} = (\mathcal{V}_1, \dots, \mathcal{V}_n)$  zusammengefasst werden. Der private Wert von Bieter  $i$  wird *unabhängig* aus  $\mathcal{V}_i$  gezogen: Bieter  $i$  hat immer die gleiche Verteilung von  $v_i$ , *egal was die anderen Bieter* an Wert bekommen.

Der Mechanismus basiert auf Verteilungen, ist aber *immer anreizkompatibel*:

Sag-die-Wahrheit ist eine dominante Strategie für jeden Bieter  $i$ , für jeden möglichen Gebote  $b_i$ , und für alle möglichen Werte  $v_{-i}$ . Die Bieter kennen ihre Verteilungen nicht (bzw. ihr Wissen darüber ändert nichts daran, dass sie immer die Wahrheit sagen wollen). Die Verteilungen sind *nur für den Entwurf und die Analyse des Mechanismus* wichtig, nicht für das strategische Verhalten der Bieter.

Die *Verteilungsfunktion*  $F_i(x)$  für Verteilung  $\mathcal{V}_i$  ist  $F_i(x) = \Pr_{v_i \sim \mathcal{V}_i}[v_i \leq x]$ . Sie hat die *Dichte*  $f_i(x)$  und es gilt  $F_i(x) = \int_{-\infty}^x f_i(x) dx$ .

**Beispiel 6.32.** *Ein-Gut-Auktion mit einem Bieter*

Mit Preis  $p$  ergibt sich der *Ertrag*  $p \cdot (1 - F_i(p))$ . Sei z.B.  $\mathcal{V}_i$  uniform auf  $[0, 1]$  verteilt, dann gilt  $F_i(x) = x$  für  $x \in [0, 1]$ . Der optimale Ertrag ist  $1/4$  für  $p = 1/2$ . ■

**Definition 6.33** (Optimaler Mechanismus). Ein *optimaler Mechanismus* ist ein anreizkompatibler Mechanismus  $(f, p_1, \dots, p_n)$ , der bei ehrlichen Geboten  $v_i = b_i$  den *erwarteten Ertrag* maximiert

$$\mathbb{E}_{v \sim \mathcal{V}} \left[ \sum_i p_i(v) \right] .$$

Anstatt die Zahlungen direkt anzuschauen, betrachten wir erst einen etwas anderen Wert.

**Definition 6.34** (Virtuelle Werte). Für Bieter  $i$ , sei  $v_i$  der Wert,  $F_i$  seine Verteilungsfunktion und  $f_i$  die Dichte der Verteilung. Dann ist der *virtuelle Wert* von Bieter  $i$

$$\varphi_i(v_i) = v_i - \frac{1 - F(v_i)}{f_i(v_i)} .$$

Es gilt immer  $v_i \geq \varphi(v_i)$ . Es kann passieren, dass  $v_i \geq 0$  und  $\varphi_i(v_i) \leq 0$ .

Intuitiv möchten wir  $v_i$  als Preis setzen, müssen aber  $(1 - F(v_i))/f_i(v_i)$  für die ehrliche Information "bezahlen".

Betrachte eine uniforme Verteilung auf  $[0,1]$ :  
Es gilt  $F(x) = x$  und  $f(x) = 1$  für  $x \in [0,1]$ . Damit ist  $\varphi(v_i) = v_i - (1 - v_i)/1 = 2v_i - 1$ .

Für jeden Bieter sind die *erwarteten* Zahlungen gleich dem *erwarteten* virtuellen Wert.

**Lemma 6.35.** *Sei  $(f, p_1, \dots, p_n)$  ein anreizkompatibler Mechanismus in einem Single-Parameter Bereich und sei  $\mathcal{V}_i$  die Verteilung von Bieter  $i$ . Dann gilt für jeden Bieter  $i$  und jedes  $v_{-i}$*

$$\mathbb{E}_{v_i \sim \mathcal{V}_i} [p_i(v_i, v_{-i})] = \mathbb{E}_{v_i \sim \mathcal{V}_i} [\varphi_i(v_i) \cdot x_i(f(v_i, v_{-i}))] .$$

*Beweis.* Die Idee ist wie folgt: Sei  $a(t) = f(t, v_{-i})$  für feste Werte  $v_{-i}$ . Wir wollen zeigen:

$$\mathbb{E}_{v_i \sim \mathcal{V}_i} [p_i(v_i, v_{-i})] = \mathbb{E}_{v_i \sim \mathcal{V}_i} [\varphi_i(v_i) \cdot x_i(a(v_i))] .$$

Dafür nutzen wir Myersons Lemma. Sei oBdA  $t_i^0 = 0$ . Dann gilt für die Zahlungen

$$\begin{aligned} p_i(v_i, v_{-i}) &= v_i \cdot x_i(a(v_i)) - \int_0^{v_i} x_i(a(t)) dt \\ &= \int_0^{v_i} t \cdot x_i'(a(t)) dt \end{aligned}$$

mit partieller Integration. Wir nehmen hier an, dass  $x$  differenzierbar ist. Wenn  $x_i$  monoton und beschränkt ist, dann folgt der Beweis ähnlich, mit einigen weiteren Argumenten und einer passenden Interpretation der Ableitung  $x_i'$ .

### Schritt 1:

Der erwartete Ertrag von Bieter  $i$  bei festen Werten  $v_{-i}$  ist

$$\begin{aligned} \mathbb{E}_{v_i \sim \mathcal{V}_i} [p_i(v_i, v_{-i})] &= \int_{z=0}^{t_i^1} p_i(z, v_{-i}) f_i(z) dz \\ &= \int_{z=0}^{t_i^1} \left[ \int_{t=0}^z t \cdot x_i'(a(t)) dt \right] f_i(z) dz . \end{aligned}$$

Die erste Gleichung nutzt Unabhängigkeit der Verteilungen – dadurch hat das feste  $v_{-i}$  keinen Einfluß auf  $\mathcal{V}_i$ .

### Schritt 2:

Jetzt müssen wir ein wenig vereinfachen. Wir vertauschen die Integrationen:

$$\begin{aligned} \int_{z=0}^{t_i^1} \left[ \int_{t=0}^z t \cdot x_i'(a(t)) dt \right] f_i(z) dz &= \int_{t=0}^{t_i^1} \left[ \int_{z=t}^{t_i^1} f_i(z) dz \right] t \cdot x_i'(a(t)) dt \\ &= \int_{t=0}^{t_i^1} (1 - F_i(t)) \cdot t \cdot x_i'(a(t)) dt , \end{aligned}$$

was den Ausdruck klarer werden lässt.

### Schritt 3:

Wir versuchen wieder partielle Integration durchzuführen. Wir nutzen:

$$g(t) = (1 - F_i(t)) \cdot t \quad \text{und} \quad h'(t) = x_i'(a(t)) .$$

Mit partieller Integration ergibt dies

$$\begin{aligned}
\mathbb{E}_{v_i \sim \mathcal{V}_i} [p_i(v_i, v_{-i})] &= (1 - F_i(t)) \cdot t \cdot x_i(a(t)) \Big|_0^{t_i^1} \\
&\quad - \int_{t=0}^{t_i^1} x_i(a(t)) \cdot (1 - F_i(t) - t \cdot f_i(t)) dt \\
&= \int_{t=0}^{t_i^1} \left( t - \frac{1 - F_i(t)}{f_i(t)} \right) \cdot x_i(a(t)) \cdot f_i(t) dt \\
&= \int_{t=0}^{t_i^1} \varphi_i(t) \cdot x_i(a(t)) \cdot f_i(t) dt \\
&= \mathbb{E}_{v_i \sim \mathcal{V}_i} [\varphi_i(t) \cdot x_i(a(v_i))] ,
\end{aligned}$$

wie erhofft. □

Für die Gesamtzahlung betrachten wir den *virtuellen Nutzen*  $\sum_i \varphi_i(v_i) \cdot x_i(f(v))$ . Aus dem Lemma folgt das zentrale Resultat: Die *erwarteten Zahlungen* sind gleich dem *erwarteten virtuellen Nutzen*.

**Satz 6.36.** Sei  $(f, p_1, \dots, p_n)$  ein anreizkompatibler Mechanismus in einem Single-Parameter Bereich und sei  $\mathcal{V}$  der Vektor der Verteilungen. Dann gilt:

$$\mathbb{E}_{v \sim \mathcal{V}} \left[ \sum_i p_i(v) \right] = \mathbb{E}_{v \sim \mathcal{V}} \left[ \sum_i \varphi_i(v_i) \cdot x_i(f(v)) \right] .$$

Daher können wir uns beim Maximieren vom Ertrag auf die Optimierung des virtuellen Nutzens konzentrieren. Das ist in vielen Fällen sehr ähnlich zur Optimierung des sozialen Nutzens.

*Beweis (Satz 6.36).* Wir nehmen die Aussage des Lemmas und nutzen den Erwartungswert über  $v_{-i}$ :

$$\begin{aligned}
\mathbb{E}_{v \sim \mathcal{V}} [p_i(v)] &= \mathbb{E}_{v_{-i} \sim \mathcal{V}_{-i}} \mathbb{E}_{v_i \sim \mathcal{V}_i} [p_i(v_i, v_{-i})] \\
&= \mathbb{E}_{v_{-i} \sim \mathcal{V}_{-i}} \mathbb{E}_{v_i \sim \mathcal{V}_i} [\varphi_i(v_i) \cdot x_i(f(v_i, v_{-i}))] \\
&= \mathbb{E}_{v \sim \mathcal{V}} [\varphi_i(v_i) \cdot x_i(f(v))] .
\end{aligned}$$

Mit der Linearität des Erwartungswertes:

$$\begin{aligned}
\mathbb{E}_{v \sim \mathcal{V}} \left[ \sum_i p_i(v) \right] &= \sum_i \mathbb{E}_{v \sim \mathcal{V}} [p_i(v)] \\
&= \sum_i \mathbb{E}_{v \sim \mathcal{V}} [\varphi_i(v_i) \cdot x_i(f(v))] \\
&= \mathbb{E}_{v \sim \mathcal{V}} \left[ \sum_i \varphi_i(v_i) \cdot x_i(f(v)) \right] . \quad \square
\end{aligned}$$

Ein optimaler anreizkompatibler Mechanismus (maximiert die Zahlungen, daher) *maximiert also den virtuellen Nutzen!*

Gilt auch umgekehrt: Ein Mechanismus, der den virtuellen Nutzen maximiert, ist ein optimaler anreizkompatibler Mechanismus?

Ja, aber nur wenn der virtuelle Nutzen eine *monotone Funktion in jedem  $v_i$  ist!* Sonst ist der Mechanismus nicht anreizkompatibel. Eine hinreichende Bedingung sind reguläre Verteilungen:

**Definition 6.37** (Reguläre Verteilung). Verteilung  $\mathcal{V}_i$  heißt *regulär*, wenn der virtuelle Wert  $\varphi_i(v_i) = v_i - \frac{1-F_i(v)}{f_i(v)}$  nicht-fallend in  $v_i$  ist.

**Korollar 6.38.** *Der optimale Mechanismus mit maximalem Ertrag in Single-Parameter Bereichen mit regulären Verteilungen  $\mathcal{V}_1, \dots, \mathcal{V}_n$  optimiert den virtuellen Nutzen der Bieter.*

Zwei Verallgemeinerungen:

– Wir nehmen nun an, die Bieter kennen alle Verteilungen und haben Gebotsstrategien. Sie geben Gebote ab abhängig von ihrem realisierten Wert und den Gebotsstrategien der anderen Bieter und deren (zufälligen) Werten. Ein Mechanismus ist *Bayes-anreizkompatibel*, wenn Sag-die-Wahrheit ein Gleichgewicht in diesem Spiel ist (ein sog. Bayes-Nash-Gleichgewicht). Auch hier gilt, dass die Maximierung des virtuellen Nutzens den optimalen Ertrag liefert. Für reguläre Verteilungen ergibt dies also sogar den optimalen Bayes-anreizkompatiblen Mechanismus.

– Für nicht-reguläre Verteilungen ist es möglich, die virtuellen Werte monoton zu machen (sog. *Bügeln*, engl: Ironing). So bekommt man den maximalen Ertrag auch für nicht-reguläre Verteilungen: Durch Optimierung des (gebügelten) virtuellen Nutzens.

**Beispiel 6.39.** *Ein-Gut-Auktion*

Wieder betrachten wir eine Ein-Gut-Auktion mit  $n$  Bietern und evtl. unterschiedlichen regulären Verteilungen:

- Das Gut geht an den Bieter mit dem *bestem virtuellen Wert*  $\max_i \varphi_i(b_i)$ . Was, wenn  $\max_i \varphi_i(b_i)$  negativ ist? Dann wird das Gut *gar nicht vergeben*.
- Der Wert  $\varphi_i^{-1}(0)$  ist also ein *Reservationspreis für Bieter  $i$* : Er muss mindestens ein Gebot  $b_i$  abgeben mit  $\varphi_i(b_i) \geq 0$ , sonst kommt er für das Gut gar nicht in Frage.
- Bekommt  $i$  das Gut, zahlt er das Maximum aus Reservationspreis und zweithöchstem Gebot, wobei das “zweithöchste Gebot” vom Bieter mit *zweithöchstem virtuellen Wert* stammt. Dieser zweithöchste virtuelle Wert muss in ein *zweithöchstes Gebot aus Sicht von  $i$*  umgerechnet werden:  $\max(\varphi_i^{-1}(0), \varphi_i^{-1}(\max_{j \neq i} \varphi_j(b_j)))$ .
- Angenommen alle  $\mathcal{V}_i$  sind z.B. gleich und uniform auf  $[0, 1]$  verteilt: Alle Funktionen  $\varphi_i(x) = 2x - 1$  sind gleich, alle Reservationspreise  $\varphi_i^{-1}(0) = 1/2$  sind gleich, und es gilt  $\varphi_i^{-1}(\varphi_j(x)) = x$ . Das Gut geht an den maximalen Bieter  $i$ , wenn sein Gebot  $b_i \geq \varphi_i^{-1}(0) = 1/2$ . Dann zahlt er  $\max(1/2, \max_{j \neq i} b_j)$ . Eine optimale Auktion ist also eine *Vickrey-Zweitpreisauktion mit Reservationspreisen*.

■

Obwohl die optimale Auktion relativ einfach aussieht, kann es schwierig werden, sie in der Praxis umzusetzen. Selbst für ein einzelnes Gut brauchen wir eventuell bis zu  $n$  verschiedene Reservationspreise und virtuelle Werte, und damit genaues Wissen über jede der Verteilungsfunktionen  $F_i$  und Dichtefunktionen  $f_i$ .

Dagegen gibt es im Fall der Ein-Gut-Auktion eine sehr viel einfachere Alternative für mehr Ertrag – mehr Wettbewerb!

Das folgende Resultat betrachtet den Ertrag für Ein-Gut-Auktionen mit gleichen regulären Verteilungen für alle Bieter. Es braucht nur einen weiteren Bieter, dann ist der Ertrag der einfachen Vickrey Auktion schon besser als der Ertrag der optimalen Auktion.

**Satz 6.40** (Bulow, Klemperer 1996). *Sei  $\mathcal{V}$  eine reguläre Verteilung und  $n \in \mathbb{N}$ . Seien  $p$  die Zahlungen der Vickrey Zweitpreisauktion mit  $n + 1$  Bieter und  $p^*$  die Zahlungen der optimalen (für  $\mathcal{V}$ ) Auktion mit  $n$  Bieter. Dann gilt*

$$\mathbb{E}_{v \sim \mathcal{V}^{n+1}} \left[ \sum_{i=1}^{n+1} p_i(v) \right] \geq \mathbb{E}_{v \sim \mathcal{V}^n} \left[ \sum_{i=1}^n p_i^*(v) \right].$$

*Beweis.* Für die Analyse betrachten wir eine *fiktive Auktion*:

1. Simuliere die optimale  $n$ -Bieter Auktion für  $\mathcal{V}$  auf Bieter  $1, \dots, n$ .
2. Wenn das Gut nicht zugewiesen wurde, gib es Bieter  $n + 1$  umsonst.

Offensichtliche Eigenschaften:

- Der erwartete Ertrag der fiktiven Auktion für  $n + 1$  Bieter ist genau der erwartete Ertrag der optimalen Auktion für  $n$  Bieter.
- Die fiktive Auktion gibt das Gut immer an einen der Bieter.

Betrachte nun die *optimale Auktion für  $n + 1$  Bieter, die immer das Gut zuweisen muss*. Sie maximiert den virtuellen Nutzen (unter der Bedingung, dass das Gut immer zugewiesen sein muss). Also gibt sie das Gut *immer an den höchsten Bieter*, selbst wenn sein *virtueller Wert negativ ist*. Die Zweitpreisauktion weist das Gut immer an den Bieter mit höchstem Wert zu.  $\mathcal{V}$  ist regulär, also ist das auch der Bieter mit dem höchsten virtuellen Wert. Das ist also *genau die optimale Auktion, die immer das Gut zuweisen muss*.

Die fiktive Auktion für  $n + 1$  Bieter muss immer das Gut zuweisen und hat den Ertrag der optimalen Auktion für  $n$  Bieter mit Verteilung  $\mathcal{V}$ .

Die Vickrey Zweitpreisauktion für  $n + 1$  Bieter muss immer das Gut zuweisen und hat den besten Ertrag (bzgl.  $\mathcal{V}$ ) aller solcher Auktionen.  $\square$

## 6.7 Zusammenfassung

Ein Mechanismus heißt anreizkompatibel, wenn “Sag die Wahrheit” eine dominante Strategie für alle Bieter darstellt. Diese Eigenschaft wird von der Vickrey-Auktion und VCG-Mechanismen erfüllt und kann direkt charakterisiert werden. Im Allgemeinen (bei mindestens drei möglichen Ergebnissen und einer unbekanntem Bewertungsfunktion) ist jeder anreizkompatible Mechanismus ein affiner Maximierer. Bei einer Einschränkung auf den Single-Parameter Bereich sind anreizkompatible Mechanismen durch Myersons Lemma definiert. Ein Problem der Anreizkompatibilität ist die Maximierung des sozialen Nutzen, wodurch Instanzen des Rucksack Problems gelöst werden müssen. Hierfür haben wir ein anreizkompatibles volles Approximationsschema betrachtet. Außerdem wurden ertragsmaximierende Mechanismen unter Verwendung des virtuellen Nutzens vorgestellt, die Geld nicht nur als Mittel zum Zweck sondern als Zielfunktion verwenden. Des Weiteren kann eine Ertragserhöhung mit einer Vickrey-Auktion und einem zusätzlichen Bieter erzielt werden. Durch das Revelationsprinzip konnten wir uns auf die Betrachtung von Mechanismen mit direkter Offenlegung beschränken.

Zur Vertiefung wird folgende Literatur empfohlen:

- Nisan, Roughgarden, Tardos, Vazirani. Algorithmic Game Theory, 2007. (Kapitel 9 und 11)
- Roughgarden. Twenty Lectures on Algorithmic Game Theory, 2016. (Kapitel 3-6)

# 7 | Online Auktionen, Sekretäre und Propheten

In vielen Anwendungen sind Kaufinteressenten nicht alle gleichzeitig in einem Markt sondern *kommen und gehen über die Zeit*. Daher betrachten wir hier Mechanismen, um ein einzelnes Gut *online* an einen Bieter zu vergeben, d.h. wenn die Bieter *einzelnd nacheinander kommen und gehen*.

Im Worst-Case ohne Kenntnis über *die Anzahl* und *den maximalen Wert* der Bieter ist es *unmöglich*, das Gut dem besten Bieter zu geben. Sogar bzgl. Approximationen des sozialen Nutzens ist jeder Mechanismus *im Worst-Case extrem schlecht*. Statt Worst-Case Analyse betrachten wir daher im Folgenden *zwei stochastische Analysemodelle* für die Ankünfte und die Werte der Bieter.

**Definition 7.1** (Online Ein-Gut Auktion).

- In Runde  $i = 1, 2, \dots$  kommt Bieter  $a_i$  und sagt einen Wert  $b_{a_i}$  für das Gut.
- Mechanismus entscheidet unmittelbar (ohne restliche Bieter  $a_{i+1}, a_{i+2}, \dots$  zu kennen), ob  $a_i$  das Gut bekommt oder nicht und was er bezahlen muss.
- Falls  $a_i$  das Gut bekommt, kann man es ihm nicht mehr wegnehmen. Falls er abgelehnt wird, kann er das Gut später nicht mehr bekommen.

## 7.1 Sekretäre und Random Order

**Definition 7.2.** Random-Order Modell

- *Werte* der Bieter sind *unbekannt*
- *Anzahl*  $n$  der Bieter ist *bekannt*
- Bieter kommen an in *uniform zufälliger Reihenfolge*

Wir wollen den *sozialen Nutzen maximieren*, d.h. das Gut dem Bieter mit höchstem Wert geben. Dies ist das *klassische Sekretärproblem*: Finde den Bieter mit höchstem Wert bei uniform zufälliger Ankunftsreihenfolge.

---

### Algorithmus 10: Sekretär-Algorithmus

---

```
Sei  $r \in \{1, \dots, n\}$  die Sample-Länge
/* Sample: */
In Runde  $i = 1, \dots, r$  sagt Bieter  $a_i$  Wert  $b_{a_i}$ . Er wird abgewiesen.
/* Akzeptanz: */
In Runde  $i = r + 1, \dots, n$  sagt Bieter  $a_i$  Wert  $b_{a_i}$ . Wenn das Gut noch nicht zugewiesen
und  $a_i$  der bisher höchste Bieter ist, dann weise  $a_i$  das Gut zu.
```

---

In diesem und allgemeineren Szenarien bewerten wir Online-Algorithmen mit dem *Wettbewerbsfaktor*:

Sei  $S^*$  eine optimale Teilmenge von Bieter, die  $\sum_{i \in S^*} b_i$  maximiert.

**Definition 7.3** (Wettbewerbsfaktor). Ein Online-Algorithmus ist  $c$ -*kompetitiv*, wenn er eine Teilmenge  $T$  von Bieter liefert mit

$$\mathbb{E} \left[ \sum_{i \in T} b_i \right] \geq \frac{1}{c} \cdot \sum_{i \in S^*} b_i .$$

$c$  ist der *Wettbewerbsfaktor* des Algorithmus.

Beachte, dass  $T$  eine zufällige Teilmenge ist. Die Randomisierung resultiert aus der zufälligen Reihenfolge der Bieter und evtl. einer internen Randomisierung des Algorithmus.

Betrachte den Algorithmus oben. Für  $r = \lfloor n/e \rfloor$  ist die Wahrscheinlichkeit, mit der der Algorithmus das Gut dem höchsten Bieter gibt, immer mindestens  $1/e$ . Daraus folgt:

**Proposition 7.4.** *Der Sekretär-Algorithmus mit  $r = \lfloor n/e \rfloor$  ist  $e$ -kompetitiv.*

Wieder soll die Auktion anreizkompatibel sein, d.h.  $b_i = v_i$ , der interne wahre Wert von Bieter  $i$ , soll eine dominante Strategie sein. Kein Bieter soll einen Anreiz zum Lügen haben, selbst wenn er

- alle Werte der anderen Bieter,
- die Ankunftsreihenfolge der Bieter,
- den Mechanismus zur Wahl des Ergebnisses und der Zahlungen, und
- mögliche interne Zufallsbits für Randomisierung im Mechanismus

kennt. Also soll der Bieter *am Ende im Nachhinein* und *für jedes mögliche Ergebnis der Randomisierungen* immer noch seinen wahren Wert mitteilen wollen.

Die Mechanismen sollen also (ex-post universell) anreizkompatibel sein.

Betrachte den Sekretär-Algorithmus. Gibt es Zahlungen, so dass daraus ein anreizkompatibler Mechanismus wird?

Dazu müsste der Algorithmus auf jedem Fall monoton sein. Daneben müssen wir unmittelbar bei der Zuweisung des Gutes die Zahlungen bestimmen, ohne die Werte der restlichen Bieter zu kennen...

Sei  $\tau = \max_{i=1, \dots, r} b_{a_i}$  der maximale Wert in der Sample-Phase. Falls der Mechanismus das Gut zuweist, dann sei  $a_{i^*}$  der Bieter, der es bekommt. Setze die Zahlungen  $p_{a_{i^*}}(b) = \tau$  und  $p_{a_i}(b) = 0$  für alle  $i \neq i^*$ .

**Proposition 7.5.** *Der Sekretär-Mechanismus ist anreizkompatibel und  $e$ -kompetitiv für die Ein-Gut-Auktion im Random-Order Modell.*

*Beweis.* Betrachte die Bieter bei fester Reihenfolge im Nachhinein.

- Kein Bieter  $a_i$  mit  $i > i^*$  oder  $i \leq r$  kann den Algorithmus durch unilaterale Änderung seines Gebotes dazu bringen, ihm das Gut zuzuweisen.
- Für Bieter  $a_i$  mit  $i = r + 1, \dots, i^*$  weist der Mechanismus das Gut dem frühesten Bieter mit  $b_{a_i} \geq \tau$  zum Preis  $\tau$  zu.

Bieter  $a_{i^*}$  mit  $v_{a_{i^*}} \geq \tau$  bekommt das Gut, zahlt  $\tau$  und ist zufrieden bei wahren Gebot. Jeder Bieter  $a_i$  mit  $i = r, \dots, i^* - 1$  hat  $v_{a_i} \leq \tau$ , bekommt und zahlt nichts und ist zufrieden bei wahren Gebot.

□

Wir verallgemeinern das Sekretärproblem, so dass  $k$  Güter versteigert werden oder z.B. nur bestimmte Ergebnisse erlaubt sind, auf Bieter-Packprobleme.

Das Bieter-Packproblem beinhaltet eine Menge  $N$  von  $n$  Bietern, wobei jeder Bieter  $i \in N$  einen privaten Wert  $v_i \geq 0$  hat. Jedes Ergebnis  $S \in A \subseteq 2^N$  ist eine *Teilmenge von Bietern*. Jeder Bieter erhält eine binäre Menge von Zeugs:  $x_i(S) = 1$ , wenn  $i \in S$ , und 0 sonst.  $A$  ergibt ein *Matroid-Packproblem*, d.h. das Paar  $(N, A)$  ist ein *Matroid* (mehr dazu später).

Beispiele:

- Ein-Gut Auktion:  $A = \{S \mid 1 \geq |S|\} = N \cup \{\emptyset\}$
- $k$ -Gut Auktion:  $A = \{S \mid k \geq |S|\}$
- Waldauktion: Jeder Bieter ist eine Kante in einem bekannten Graphen  $G$ . Die Auktion kann eine kreisfreie Teilmenge von Kanten/Bietern wählen  
 $A = \{S \mid S \text{ ist kreisfreie Menge von Kanten in } G\}$ .

Bei einem *Online-Mechanismus* sind  $N$  und  $A$  am Anfang bekannt. Die Bieter erscheinen in zufälliger Reihenfolge, und teilen bei ihrer Ankunft den privaten Wert mit. Der Online-Mechanismus entscheidet unmittelbar und endgültig nach der Ankunft eines Bieters: Soll der Mechanismus den Bieter *auswählen* oder *ablehnen*? Wenn er ausgewählt wird, was *zahlt* der Bieter? Wenn er abgelehnt wird, ist seine Zahlung 0.

Jeder Bieter wird entschieden bevor weitere Bieter ankommen. Die Entscheidung über die Auswahl, Ablehnung und Zahlung ist endgültig.

Ziele für den Entwurf von Online-Mechanismen:

1. ex-post universell anreizkompatibel
2. approximiere sozialen Nutzen  $\sum_{i \in S} b_i$ , möglichst kleiner Wettbewerbsfaktor
3. berechenbar in polynomieller Zeit

Wir betrachten zuerst Online-Mechanismen für einen graphischen Matroid:

Bei einem Graphischen Matroid sind die Bieter Kanten eines *zusammenhängenden Graphen*  $G = (K, N)$ . Die Ergebnismenge  $A$  enthält alle *Wälder* (*kreisfreie Mengen von Kanten*), dabei ist jedes  $S \in A$  mit *maximaler Kardinalität ist ein Spannbaum* von  $G$ . Jeder Spannbaum besteht aus  $k = |K| - 1$  Kanten. Alle Bieter  $i$  haben den privaten Wert  $b_i$ , wir schreiben  $v(S) = \sum_{i \in S} v_i$  und  $b(S) = \sum_{i \in S} b_i$ .

Der Greedy-Algorithmus von Kruskal berechnet einen optimalen Spannbaum mit maximaler Gesamtbewertung.

Die Austauschenschaft wird sich als sehr wichtig herausstellen:

Sei  $S$  Spannbaum und  $S^*$  optimaler Spannbaum. Es gibt Paare von Bietern  $(a_1, a_1^*), \dots, (a_k, a_k^*) \in S \times S^*$  so dass für alle  $1 \leq i \leq k$ .

- $S_i = (S \cup \{a_1^*, \dots, a_i^*\}) \setminus \{a_1, \dots, a_i\}$  ist Spannbaum
- $S^* = S_k$
- $b(S_i) \geq b(S_{i-1})$

Wie sieht ein guter Online-Mechanismus für eine Waldauktion aus?

---

**Algorithmus 11:** Random-Threshold

---

Setze  $p_i(b) \leftarrow 0$  für alle Bieter  $i \in N$   
 Lehne die ersten  $n/2$  Bieter ab, sei dies die Menge  $Y$   
 Wähle  $j \in \{0, 1, 2, \dots, \lceil \log k \rceil\}$  uniform zufällig  
 Setze Schranke  $\tau \leftarrow \max_{x \in Y} b_x / 2^j$ , setze  $S \leftarrow \emptyset$   
**for** jede Runde  $i = n/2 + 1, \dots, n$  **do**  
   Sei  $a_i$  der Bieter, der in Runde  $i$  ankommt.  
   **if**  $b_{a_i} \geq \tau$  und  $(S \cup \{a_i\}) \in A$  **then**  
      $S \leftarrow S \cup \{a_i\}$  und  $p_{a_i}(b) \leftarrow \tau$

---

**Satz 7.6.** *Der Random-Threshold Mechanismus ist anreizkompatibel und  $O(\log k)$ -kompetitiv für Waldauktionen.*

*Beweis.* Die Argumente zur Anreizkompatibilität sind sehr ähnlich zum Sekretär-Mechanismus (Übung). Wir zeigen hier den Wettbewerbsfaktor.

Wir konzentrieren uns auf Bieter aus  $S^*$  mit signifikantem Wert.

Sei  $S^*$  ein optimaler Spannbaum. Wir nummerieren die  $k$  Bieter aus  $S^*$  mit  $1, \dots, k$  in absteigender Reihenfolge der Werte  $b_1 \geq \dots \geq b_k$ . Beachte, dass  $b_1$  der höchste Bieter aus  $N$  (Greedy-Algorithmus) ist, aber die anderen nicht unbedingt die nächsten  $k - 1$  Bieter mit den höchsten Werten aus  $N$  sind.

Wähle  $q$ , so dass gilt:  $b_q \geq b_1/k$  und entweder  $q = k$  oder  $b_{q+1} < b_1/k$ . Es ist zu beachten, dass

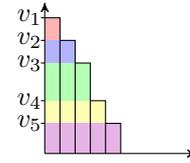
$$\sum_{i=q+1}^k b_i < b_1, \quad \text{daher} \quad \sum_{i=1}^q b_i \geq b(S^*)/2.$$

Die  $q$  höchsten Bieter in  $S^*$  liefern also mindestens die Hälfte des optimalen sozialen Nutzens.

Die Analyse nutzt Wertklassen basierend auf den Bietern  $i = 1, \dots, k$  aus  $S^*$ . O.B.d.A. nehmen wir an, dass  $b_1 > b_2 > \dots > b_k$ .

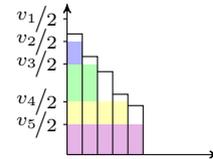
Es gibt genau  $i$  Bieter in  $S^*$  mit Wert mindestens  $v_i$ . Sei  $m_i(T)$  die Anzahl Bieter in  $T \subset N$  mit Wert mindestens  $b_i/2$ . Wir sehen, dass

$$\sum_{i=1}^q b_i = \left[ \sum_{i=1}^{q-1} (b_i - b_{i+1}) \cdot i \right] + b_q \cdot q.$$



Für jede Menge  $T$  ergibt sich

$$b(T) \geq \frac{1}{2} \cdot \left[ \sum_{i=1}^{q-1} (b_i - b_{i+1}) \cdot m_i(T) \right] + \frac{1}{2} \cdot b_q \cdot m_q(T).$$



**Lemma 7.7.** *Sei  $S$  die Ergebnismenge von Random-Threshold. Für jedes  $i = 1, \dots, q$  gilt*

$$\mathbb{E}[m_i(S)] \geq \frac{1}{8(\lceil \log k \rceil + 1)} \cdot i.$$

Mit diesem Lemma folgt der Satz:

$$\begin{aligned}
\mathbb{E}[b(S)] &\geq \frac{1}{2} \cdot \left[ \sum_{i=1}^{q-1} (b_i - b_{i+1}) \cdot \mathbb{E}[m_i(S)] \right] + \frac{1}{2} \cdot b_q \cdot \mathbb{E}[m_q(S)] \\
&\geq \frac{1}{16(\lceil \log k \rceil + 1)} \left[ \sum_{i=1}^{q-1} (b_i - b_{i+1}) \cdot i \right] + \frac{1}{16(\lceil \log k \rceil + 1)} \cdot b_q \cdot q \\
&= \frac{1}{16(\lceil \log k \rceil + 1)} \cdot \sum_{i=1}^q b_i \\
&\geq \frac{1}{32(\lceil \log k \rceil + 1)} \cdot b(S^*) .
\end{aligned}$$

□

*Beweis (Lemma 7.7).* Wir zeigen das Lemma für jeden Wert von  $i$  induktiv. Der Induktionsanfang mit  $i = 1$  ist einfach (Übung). Betrachten wir also die Fälle  $1 < i \leq q$ .

Sei  $a^*$  der Bieter mit maximalem Wert. Wir bedingen auf ein Ereignis  $E$ , dass zwei Annahmen gleichzeitig eintreten:

- (1) Der höchste Bieter ist im Sample  $a^* \in Y$ , und
- (2) für  $\tau$  wird  $j$  so gewählt, dass  $b_i \geq b_{a^*}/2^j \geq b_i/2$ .

Wir können  $S^*$  mit dem Greedy-Algorithmus berechnen. Daher ist  $b_1 = b_{a^*}$  und  $b_q \geq b_1/k \geq b_{a^*}/2^{\lceil \log k \rceil}$ . Somit gibt es ein passendes  $j$  für Annahme (2) in jedem Fall  $1 < i \leq q$ .

Der Algorithmus wählt dieses passende  $j$  mit Wahrscheinlichkeit  $1/(\lceil \log k \rceil + 1)$ . Die gesamte Wahrscheinlichkeit von Ereignis  $E$  ist damit  $\Pr[E] = 1/(2(\lceil \log k \rceil + 1))$ .

Wir zeigen nun eine Schranke unter der Bedingung, dass Ereignis  $E$  eintritt.

Die höchsten  $i$  Bieter aus  $S^*$  bilden eine kreisfreie Menge  $S' = \{1, \dots, i\}$ . Nach Annahme (2) sind alle Werte  $b_1 \geq \dots \geq b_i \geq \tau = b_{a^*}/2^j$ . Mit Annahme (1) ist  $a^* = 1$  in  $Y$ . Daher sind im Erwartungswert mindestens  $(i-1)/2 \geq i/4$  Bieter von  $S'$  nicht in  $Y$  und können vom Algorithmus gewählt werden. Es gilt also  $\mathbb{E}[|S' \setminus Y| \mid E] \geq i/4$ . Wegen der Austauschbarkeit wird der Algorithmus in diesem Fall auch mindestens  $S' \setminus Y$  viele Bieter auswählen. Die erwartete Größe der Ausgabe  $S$  des Algorithmus bedingt auf Ereignis  $E$ , ist also

$$\mathbb{E}[|S| \mid E] \geq \mathbb{E}[|S' \setminus Y| \mid E] \geq i/4.$$

Da  $\tau \geq b_i/2$  und jeder gewählte Bieter einen Wert mindestens  $\tau$  hat:

$$\mathbb{E}[m_i(S) \mid E] = \mathbb{E}[|S| \mid E] \geq i/4.$$

Wir entfernen die Bedingung auf Ereignis  $E$ , indem wir mit  $\Pr[E]$  multiplizieren. □

Der Algorithmus und die Analyse können direkt auf beliebige Matroidauktionen angewendet werden.

**Definition 7.8 (Matroid).** Ein Tupel  $M = (N, A)$  ist ein *Matroid*, wenn  $N = \{1, \dots, n\}$  eine endliche Menge von Bietern und  $A$  eine nicht-leere Familie von Teilmengen von  $N$  ist, für die gilt:

- Wenn  $I \in A$  und  $J \subseteq I$ , dann ist auch  $J \in A$ , und
- wenn  $I, J \in A$  und  $|J| < |I|$ , dann gibt es ein  $i \in I \setminus J$  mit  $J \cup \{i\} \in A$ .

Für Matroide verwenden wir die Notation:

- Eine Menge  $I \in A$  heißt *unabhängig*.
- Eine maximale unabhängige Menge  $B \in A$  heißt *Basis*.
- Die Kardinalität jeder Basis ist gleich und heißt *Rang*  $rk(M)$  des Matroids. Sei  $k = rk(M)$ .

**Definition 7.9** (Gewichteter Matroid). • Ein Matroid mit Gewichten  $v_i \in \mathbb{R}$  für alle  $i \in N$  heißt *gewichtet*.

- Das Gewicht einer unabhängigen Menge  $I$  ist  $v(I) = \sum_{i \in I} v_i$ .
- Eine *optimale Basis* ist eine Basis mit maximalem Gewicht.

Zur Vollständigkeit:

Der Greedy-Algorithmus von Kruskal berechnet eine optimale Basis mit maximalem Gesamtgewicht.

Austauscheigenschaft: Sei  $Y$  Basis und  $Y^*$  optimale Basis. Es gibt eine Folge von Paaren  $(y_1, y_1^*), \dots, (y_k, y_k^*) \in Y \times Y^*$ , so dass für  $1 \leq i \leq k$

- $Y_i = (Y \cup \{y_1^*, \dots, y_i^*\}) \setminus \{y_1, \dots, y_i\}$  ist eine Basis
- $Y^* = Y_k$
- $v(Y_i) \geq v(Y_{i-1})$

**Satz 7.10** (Babaioff, Immorlica, Kleinberg 2007). *Random-Threshold ist anreizkompatibel und  $O(\log k)$ -kompetitiv für beliebige Matroidauktionen.*

Matroid-Sekretär Vermutung: Für jeden Matroid gibt es einen Algorithmus, der im Random-Order Modell ...

**Schwach:** ... konstant-kompetitiv ist.

**Stark:** ...  $e$ -kompetitiv ist.

Die Vermutung wurde in den letzten zehn Jahren für viele Klassen von Matroiden bewiesen. Für den allgemeinen Fall ist sie allerdings weiterhin offen. Die momentan besten Algorithmen lieferten Lachish (2014) und Feldman, Svensson und Zenklusen (2018). Sie erreichen einen Wettbewerbsfaktor von  $O(\log \log k)$ . Diese Algorithmen ergeben allerdings nicht unbedingt anreizkompatible Mechanismen.

Für graphische Matroide beweisen wir hier die schwache Vermutung.

---

**Algorithmus 12:** Parallele Sekretäre

---

Fixiere beliebige Reihenfolge  $k_1, k_2, k_3, \dots$  der *Knoten des Graphen*

Wähle  $X \in \{0, 1\}$  uniform zufällig

**if**  $X = 1$  **then**

  └ Orientiere jede Kante  $e \in E$  zum Knoten mit kleinerem Index

**else**

  └ orientiere jede Kante zum Knoten mit größerem Index

**for** jeden Knoten  $k_i$  **parallel do**

  └ Führe den Sekretär-Algorithmus aus auf den eingehenden Kanten und wähle damit maximal eine eingehende Kante von  $k_i$  aus. Setze Zahlungen wie im Sekretär-Mechanismus für die Ein-Gut-Auktion

---

**Satz 7.11** (Korula, Pal 2009). *Parallele-Sekretäre ist anreizkompatibel und  $2e$ -kompetitiv für Waldauktionen.*

*Beweis.* Anreizkompatibilität folgt wie beim Sekretär-Mechanismus (Übung).

Wir betrachten hier den Wettbewerbsfaktor.

Durch die Orientierung wird die (direkte oder inverse) Sortierung der Knoten zu einer topologischen Sortierung.  $G$  wird also in jedem Fall zu einem gerichteten, azyklischen Graphen. Wenn jeder Knoten eine beliebige eingehende Kante wählt, bleibt  $S$  in jedem Fall kreisfrei. Wir müssen also nur noch  $\mathbb{E}[b(S)]$  beschränken. Sei  $G_X$  der gerichtete Graph für  $X \in \{0, 1\}$ , und sei  $h_X(k_i)$  eine eingehende Kante mit höchstem Wert von  $k_i$  in  $G_X$ . Sei  $S_X = \{h_X(k_i) \mid k_i \in K\}$ , und  $S^*$  ein optimaler Wald in  $G$ .

**Proposition 7.12.**

$$b(S^*) \leq \sum_{k_i \in K} b_{h_0(k_i)} + b_{h_1(k_i)} = b(S_0) + b(S_1) .$$

Bedingt auf die Wahl von  $X$  liefert der Algorithmus für Knoten  $k_i$  im Erwartungswert mindestens  $1/e$  des höchsten Wertes einer eingehenden Kante von  $k_i$ . Daher ergibt sich sowohl für  $x = 0$  als auch  $x = 1$

$$\mathbb{E}[b(S) \mid X = x] \geq 1/e \cdot b(S_x) .$$

Mit der Proposition erkennen wir

$$\begin{aligned} \mathbb{E}[b(S)] &= \frac{1}{2} \cdot (\mathbb{E}[b(S) \mid X = 0] + \mathbb{E}[b(S) \mid X = 1]) \\ &\geq \frac{1}{2e} \cdot (b(S_0) + b(S_1)) \\ &\geq \frac{1}{2e} \cdot b(S^*) . \end{aligned}$$

□

## 7.2 Prophetische Ungleichung

Hier wird nun der private Wert  $v_i$  für jeden Bieter  $i$  unabhängig aus einer *bekannt* Verteilung  $\mathcal{V}_i$  mit *nicht-negativen Werten* und *endlichem Erwartungswert* gezogen.

**Definition 7.13.** Verteilungsmodell

- Werte der Bieter und *Ankunftsreihenfolge* sind *unbekannt*
- *Anzahl*  $n$  der Bieter ist *bekannt*
- Wert  $v_i$  wird unabhängig gezogen aus *bekannter Verteilung*  $\mathcal{V}_i$  für Bieter  $i$ , Bieter bietet bei Ankunft  $b_i$

Wir betrachten zuerst die Ein-Gut-Auktion und wollen *sozialen Nutzen maximieren*. Später betrachten wir auch noch *Ertragsmaximierung*.

---

**Algorithmus 13:** Prophet

---

Sei  $p_i(b) \leftarrow 0$  für alle Bieter  $i \in N$

Sei  $\tau \leftarrow \mathbb{E}_{x_j \sim \mathcal{V}_j} [\max_j x_j] / 2 = \mathbb{E}[v_{\max}] / 2$

**for** jede Runde  $i = 1, 2, \dots, n$  **do**

**if**  $b_{a_i} \geq \tau$  und Gut noch verfügbar **then**  
└─ Gib Bieter  $a_i$  das Gut und setze  $p_{a_i}(b) \leftarrow \tau$

---

**Satz 7.14** (Krengel, Sucheston 1978). *Der Prophet Mechanismus ist anreizkompatibel und 2-kompetitiv für die Ein-Gut-Auktion im Verteilungsmodell.*

*Beweis.* Anreizkompatibilität folgt wieder sehr ähnlich wie beim Sekretär-Mechanismus (Übung). Wir beschränken den Wettbewerbsfaktor und nehmen an, dass  $b_i = v_i$  für alle Bieter  $i$ .

Der Algorithmus vergibt das Gut in der ersten Runde  $i^*$ , in der ein Bieter mit Wert  $v_{a_{i^*}} \geq \tau$  erscheint. Wenn kein solcher Bieter kommt, vergibt er das Gut gar nicht, dann sei  $v_{a_{i^*}} = 0$ . Wir werden zeigen, dass im Erwartungswert  $\mathbb{E}[v_{a_{i^*}}] \geq \tau$ . Da alle Werte aus Verteilungen gezogen werden, ist auch das Optimum  $v_{\max}$  eine Zufallsvariable. Wir werden also zeigen:

$$\mathbb{E}[v_{a_{i^*}}] \geq \frac{1}{2} \cdot \mathbb{E}_{x_j \sim \nu_j}[\max_j x_j] = \frac{1}{2} \cdot \mathbb{E}[v_{\max}].$$

Sei  $q(x) = \Pr[v_{\max} \geq x]$  die Wahrscheinlichkeit, dass mindestens ein Bieter einen Wert mindestens  $x$  hat. Wenn niemand Wert mindestens  $\tau$  hat, dann wird das Gut in allen Runden nicht vergeben. Das passiert mit Wahrscheinlichkeit  $(1 - q(\tau))$ . Für jeden Bieter  $a_i$  gilt also: Mit Wahrscheinlichkeit mindestens  $(1 - q(\tau))$  ist das Gut in Runde  $i$  noch verfügbar. Mit Wahrscheinlichkeit  $q_i(x) = \Pr[v_{a_i} > x]$  hat Bieter  $a_i$  einen Wert  $v_{a_i} > x \geq \tau$ . Dann gibt der Algorithmus das Gut an  $a_i$ . Betrachte das Ereignis  $E_i(x)$ , dass (1) der Algorithmus das Gut an  $a_i$  gibt und (2) dessen Wert  $v_{a_i} > x$  ist. Es gilt

$$\Pr[E_i(x)] \geq (1 - q(\tau)) \cdot q_i(x).$$

Für jedes  $x \geq \tau$  kann das Gut nur an maximal einen der Bieter mit Wert mindestens  $x$  vergeben werden. Daher kann nur maximal eines der Ereignisse  $E_i(x)$  eintreten und es gilt:

$$\Pr[v_{a_{i^*}} > x] = \sum_{i=1}^n \Pr[E_i(x)] \geq (1 - q(\tau)) \cdot \sum_{i=1}^n q_i(x).$$

Es können durchaus mehrere Bieter gleichzeitig Wert mindestens  $x$  haben. Also gilt für die Wahrscheinlichkeit, dass mindestens einer Wert mindestens  $x \geq \tau$  hat: (Union Bound)

$$\Pr[v_{\max} > x] \leq \sum_{i=1}^n q_i(x).$$

Also gilt für  $x \geq \tau$  eine Beziehung zwischen Algorithmus und Optimum:

$$\Pr[v_{a_{i^*}} > x] \geq (1 - q(\tau)) \cdot \Pr[v_{\max} > x].$$

Für die Erwartungswerte nutzen wir die Integral-Definition für Verteilungen über nicht-negative Werte:

$$\begin{aligned} \mathbb{E}[v_{a_{i^*}}] &= \int_{x=0}^{\infty} \Pr[v_{a_{i^*}} > x] dx \\ &= q(\tau) \cdot \tau + \int_{x=\tau}^{\infty} \Pr[v_{a_{i^*}} > x] dx \\ &\geq q(\tau) \cdot \tau + (1 - q(\tau)) \int_{x=\tau}^{\infty} \Pr[v_{\max} > x] dx \\ &= q(\tau) \cdot \tau + (1 - q(\tau)) \left( \mathbb{E}[v_{\max}] - \int_{x=0}^{\tau} \Pr[v_{\max} > x] dx \right) \\ &\geq q(\tau) \cdot \tau + (1 - q(\tau)) (\mathbb{E}[v_{\max}] - \tau). \end{aligned}$$

Hier wird die natürliche Abwägung deutlich: Je höher  $\tau$ , desto mehr Wert erreichen wir bei Vergabe des Gutes, aber desto kleiner ist die Wahrscheinlichkeit  $q(\tau)$ , dass überhaupt jemand existiert, der Wert mind.  $\tau$  für das Gut hat.

Für  $\tau = \mathbb{E}[v_{\max}]/2$  fallen die  $q(\tau)$ -Terme weg, und wir erhalten  $\mathbb{E}[v_{a_{i^*}}] \geq 1/2 \cdot \mathbb{E}[v_{\max}]$  wie gewünscht.  $\square$

Die Ungleichung

$$\mathbb{E}[v_{a_i^*}] \geq \frac{1}{2} \cdot \mathbb{E}[v_{\max}]$$

heißt prophetische Ungleichung (prophet inequality), da sie den Wert gegenüber einem optimalen Propheten abschätzt, der die genaue Eingabe kennt.

Wenn wir den Algorithmus auf virtuelle Werte anwenden anstatt auf die originalen Werte, erhalten wir damit eine prophetische Ungleichung für den erwarteten virtuellen Nutzen. Damit erhalten wir eine 2-Approximation des erwarteten Ertrags der ertragsoptimalen anreizkompatiblen (Offline-)Auktion.

---

**Algorithmus 14:** Ertragsprophet

---

Sei  $p_i(b) \leftarrow 0$  für alle  $i \in N$

Wir schreiben  $(x)^+ = \max(0, x)$

Setze  $\tau \leftarrow \frac{1}{2} \cdot \mathbb{E}_{v \sim \mathcal{V}}[\max_i(\varphi_i(v_i))^+]$

**for** jede Runde  $i = 1, 2, \dots, n$  **do**

**if**  $\varphi_{a_i}(b_{a_i}) \geq \tau$  und Gut noch verfügbar **then**  
        └ Gib  $a_i$  das Gut und setze  $p_{a_i}(b) \leftarrow \varphi_{a_i}^{-1}(\tau)$

---

**Satz 7.15.** *Ertragsprophet ist anreizkompatibel für reguläre Verteilungen und 2-kompetitiv bzgl. der ertragsoptimalen Ein-Gut-Auktion.*

*Beweis.* Anreizkompatibilität folgt wie oben, da für reguläre Verteilungen  $\varphi_i(v_i)$  monoton steigend in  $v_i$  ist. Die Analyse des Faktors oben kann mit minimalen Änderungen genauso für virtuelle Werte und virtuellen sozialen Nutzen durchgeführt werden.

Daraus folgt also, dass

$$\mathbb{E}[\varphi_{a_i^*}(v_{a_i^*})] \geq \frac{1}{2} \cdot \mathbb{E}[\max_i(\varphi_i(v_i))^+]$$

Beachte, dass  $\mathbb{E}[\max_i(\varphi_i(v_i))^+]$  der virtuelle Wert ist, den man erzielt, wenn das Gut immer dem Bieter mit höchstem nicht-negativem virtuellen Wert gegeben wird. Dies ist also der virtuelle Wert der ertragsoptimalen (Offline-)Auktion. Die erwarteten Zahlungen sind der erwartete virtuelle Wert, also gilt

$$\mathbb{E}[p_{a_i^*}(v)] \geq \frac{1}{2} \cdot \mathbb{E}\left[\sum_i p_i^*(v)\right],$$

wobei  $p_i^*$  die Zahlungen der ertragsoptimalen (Offline-)Auktion sind. □

Man bekommt auch für die Ertragsmaximierung im Offline-Fall einen einfachen anreizkompatiblen Mechanismus: Berechne  $\tau$  und biete den Bietern das Gut nacheinander in beliebiger Reihenfolge zum Preis  $\tau$  zum Kauf an, solange bis einer das Gut kauft. Damit erzielen wir mindestens die Hälfte des Ertrags jeder anderen anreizkompatiblen Auktion.

Wir können das Szenario nun kanonisch auf Matroidauktionen erweitern:

- Bekannt: Menge  $N$  von Bietern, Ergebnismenge  $A$ , Matroid  $M = (N, A)$
- Bieter  $i \in N$  hat unbekanntem privaten Wert  $v_i \sim \mathcal{V}_i$ , Verteilung  $\mathcal{V}_i$  bekannt
- Bieter kommen in unbekannter Ordnung an. Online-Mechanismus entscheidet unmittelbar: Auswahl/Ablehnung und Zahlungen. Entscheidungen sind endgültig
- Ziel: Erstelle eine unabhängige Menge  $S \in A$  mit maximalem Gesamtwert. Mechanismus soll anreizkompatibel sein.

Wir analysieren hier eine Klasse von Mechanismen, die – anstatt einer globalen Schranke  $\tau$  – *deterministische Schranken*  $\tau_i$  für jeden Bieter  $i \in N$  setzen. Sei  $\tau_i = \infty$ , wenn  $S \cup \{i\} \notin A$ . Der Algorithmus *akzeptiert einen Bieter genau dann, wenn  $b_i \geq \tau_i$* . In diesem Fall *zahlt Bieter  $i$  dann  $\tau_i$* .

Für so einen Ansatz brauchen wir eine sorgfältige Ausgestaltung der  $\tau_i$ . Insbesondere möchten wir, dass die Schranken dazu führen, dass (1) nicht zu viele Bieter mit kleinem Wert ausgewählt und (2) nicht zu viele wertvolle Bieter abgelehnt werden. Die Idee der  $\alpha$ -*balancierten* Schranken formalisiert diese Bedingungen.

Für jeden Bieter  $i \in N$ , sei  $v_i = b_i$  der private Wert und  $v'_i$  ein beliebiger gesampter Wert. Beide werden unabhängig aus  $\mathcal{V}_i$  gezogen. Die Eingabefolge sei  $\sigma = (a_1, v_{a_1}), \dots, (a_n, v_{a_n})$ . In der nachfolgenden Definition der Schranken unten fixieren wir die Folge (und damit alle  $v_i$ 's) und fordern eine Bedingung für jede solche Folge. Die ausgewählte Menge von Bietern nennen wir  $S = S(\sigma)$ . Die optimale Basis für die gesampleten Werte  $v'$  sei  $B'$ . Mit der Austauschenschaft gibt es mindestens eine Aufteilung von  $B'$  in  $B_c$  und  $B_r$ , so dass  $S \cup B_r$  eine Basis von  $M$  darstellt. Von allen solchen Aufteilungen sei  $(B_c(S), B_r(S))$  diejenige, die den gesampleten Gesamtwert  $v'(B_r(S))$  maximiert.

**Definition 7.16** ( $\alpha$ -balancierte Schranke). Sei  $\alpha > 0$ . Der Mechanismus hat  $\alpha$ -*balancierte Schranken*, wenn für jede Eingabefolge  $\sigma$  und  $X$  disjunkt von  $S = S(\sigma)$  mit  $S \cup X \in A$  gilt, dass die deterministischen Schranken  $\tau_i = \tau_i(\sigma)$  folgende Bedingungen erfüllen:

$$\sum_{i \in S} \tau_i \geq \left(\frac{1}{\alpha}\right) \cdot \mathbb{E}_{v' \sim \mathcal{V}}[v'(B_c(S))] \quad (7.1)$$

$$\sum_{i \in X} \tau_i \leq \left(1 - \frac{1}{\alpha}\right) \cdot \mathbb{E}_{v' \sim \mathcal{V}}[v'(B_r(S))] . \quad (7.2)$$

**Proposition 7.17.** *Wenn der Mechanismus  $\alpha$ -balancierte Schranken hat, dann ist er anreizkompatibel und  $\alpha$ -kompetitiv für Matroidauktionen.*

---

**Algorithmus 15:** Expected-Margin-Thresholds
 

---

Sei  $p_i(b) \leftarrow 0$  für alle  $i \in N$ ,  $S \leftarrow \emptyset$

**for** jede Runde  $i = 1, 2, \dots, n$  **do**

**if**  $(S \cup \{a_i\}) \notin A$  **then**

$\tau_i \leftarrow \infty$

**else**

$$\tau_i \leftarrow \frac{1}{2} \cdot \mathbb{E}_{v' \sim \mathcal{V}}[v'(B_r(S)) - v'(B_r(S) \cup \{a_i\})] ,$$

        wobei alle  $v'_i \sim \mathcal{V}_i$  unabhängig gezogen.

**if**  $b_{a_i} \geq \tau_i$  **then**

$S \leftarrow S \cup \{a_i\}$  und  $p_{a_i}(b) \leftarrow \tau_i$

---

**Satz 7.18** (Kleinberg, Weinberg 2012). *Expected-Margin-Thresholds hat 2-balancierte Schranken, ist anreizkompatibel und 2-kompetitiv für Matroidauktionen.*

### 7.3 Zusammenfassung

Das Verteilungsmodell erscheint “netter” als das Random-Order Modell:

Es erlaubt bessere Faktoren in der Ein-Gut-Auktion (2 vs.  $e$ ). Für Matroidauktionen ist die Verbesserung drastisch (2 vs.  $\omega(1)$ ), hier hängt die abschließende Bewertung aber auch von der Lösung der Matroid-Sekretär Vermutung ab. Für eine Reihe von weiteren Packungsproblemen hat man in den

letzten Jahren Algorithmen im Random-Order und im Verteilungsmodell erforscht. So existieren mittlerweile Ansätze für Online-Varianten von Rucksack, Matching, Independent Set, Packungs-Integer-Programs und weiteren wichtigen Problemstellungen. Die Garantien für Algorithmen in stochastischen Online-Modellen sind aber oftmals nur ansatzweise verstanden und weiterhin ein interessantes Gebiet aktueller Forschung.

Zur Vertiefung wird folgende Literatur empfohlen:

- M. Babaioff, N. Immorlica, R. Kleinberg. Matroids, Secretary Problems, and Online Mechanisms. SODA 2007.
- N. Korula, M. Pal. Algorithms for Secretary Problems on Graphs and Hypergraphs. ICALP 2009.
- O. Lachish.  $O(\log \log \text{rank})$ -Competitive Ratio for the Matroid Secretary Problem. FOCS 2014.
- M. Feldman, O. Svensson, R. Zenklus. A Simple  $O(\log \log \text{rank})$ - Competitive Algorithm for the Matroid Secretary Problem. Math. Oper. Res. 43(2):638-650, 2018.
- P. Freeman. The Secretary Problem and its Extensions: A Review. Intl. Stat. Rev. 51(2):189-206, 1983.
- R. Kleinberg, M. Weinberg. Matroid Prophet Inequalities. STOC 2012.
- U. Krengel, L. Sucheston. On semiamarts, amarts, and processes with finite value. Adv. in Prob. Related Topics 4:197-266, 1978.



# 8 | Mechanismen als Spiele

Bisher haben wir anreizkompatible Mechanismen betrachtet, bei denen jeder Bieter  $i$  seinen optimalen Nutzen durch ein wahres Gebot  $b_i = v_i$  erreichen konnte. Wenn wir diese Eigenschaft nicht fordern, werden die Bieter lügen, um ihren Nutzen zu maximieren. Eine solche Auktion können wir als Spiel auffassen, bei dem die Gebote Strategien entsprechen. Wie werden sich die Bieter in solch einem Spiel verhalten? Welche Mechanismen können wir ohne die Anforderung der Anreizkompatibilität entwerfen?

## 8.1 Generalized-Second-Price Auktionen

Wir modellieren ein bekanntes Beispiel für solche Auktionen, die Zuweisung von Werbeslots auf einer Seite mit Suchresultaten.

- $n$  Werbekunden/Spieler, Spieler  $i$  hat *Bewertung*  $v_i \geq 0$  *pro Click*
- $n$  Werbeslots auf der Seite (falls weniger: unsichtbare Dummy-Slots)
- Slot  $j$  hat *Clickrate*  $\alpha_j \geq 0$
- Spieler  $i$  hat *Relevanz*  $\gamma_i \geq 0$  zum Suchbegriff
- Wenn Spieler  $i$ 's Anzeige in Slot  $j$  erscheint, erhält er  $\alpha_j \cdot \gamma_i$  Clicks.

Der Mechanismus sammelt Gebote  $b_i$  ein, weist Werbeanzeigen auf Slots zu und setzt Zahlungen  $p_i$  pro Click.

Der Nutzen von Spieler  $i$ , wenn er Slot  $j$  erhält, ist

$$\alpha_j \cdot \gamma_i \cdot (v_i - p_i) .$$

Wir *nehmen an*  $\gamma_i = 1$ . Wir erhalten trotzdem die gleichen Resultate wie im allgemeinen Fall!

Als Beispiel betrachten wir zuerst den VCG-Mechanismus in diesem Szenario.

VCG maximiert den sozialen Nutzen:  $\sum_{i=1}^n \alpha_j b_i$  mit Spieler  $i$  in Slot  $j_i$ . Die Spieler und Slots werden geordnet, so dass  $b_1 \geq \dots \geq b_n$  und  $\alpha_1 \geq \dots \geq \alpha_n$ . Mit einem einfachen Austausch-Argument folgt, dass der soziale Nutzen maximal ist bei  $j_i = i$ . Die VCG-Zahlungen mit Clarke-Regel entsprechen:

$$\begin{aligned} \alpha_i p_i &= h_i(b_{-i}) - \sum_{j \neq i} \alpha_j b_j = \sum_{j=1}^{i-1} \alpha_j b_j + \sum_{j=i}^{n-1} \alpha_j b_{j+1} - \sum_{j \neq i} \alpha_j b_j \\ &= \sum_{j=i+1}^n b_j (\alpha_{j-1} - \alpha_j) = b_{i+1} (\alpha_i - \alpha_{i+1}) + \alpha_{i+1} p_{i+1} . \end{aligned}$$

Einsetzen ergibt den Nutzen von Spieler  $i$  als  $\alpha_i (v_i - b_{i+1}) - \alpha_{i+1} (b_{i+1} - p_{i+1})$ .

VCG hat viele gute Eigenschaften, es ist IC und kann sehr schnell berechnet werden. Dennoch wird VCG heute von keiner großen Suchmaschine benutzt. Stattdessen nutzen alle Varianten des folgenden Schemas, bekannt als die Generalized-Second-Price Auktion (GSP):

- Allokation maximiert sozialen Nutzen, i.e., Spieler  $i$  bekommt Slot  $i$ .
- Zahlung von Spieler  $i$  ist nächsthöheres Gebot  $b_{i+1}$ . Hier sei  $b_{n+1} = 0$ .
- Nutzen von Spieler  $i$  in GSP ist  $\alpha_i(v_i - b_{i+1})$ .

Beachte, dass GSP für die gleiche Menge an Geboten höhere Zahlungen setzt:

$$\sum_{j=i+1}^n b_j(\alpha_{j-1} - \alpha_j) \leq b_{i+1} \sum_{j=i+1}^n (\alpha_{j-1} - \alpha_j) \leq \alpha_i b_{i+1} .$$

Allerdings ist unklar, ob die Spieler gleiche Gebote bei VCG und GSP abgeben. Es stellt sich heraus, dass GSP *nicht anreizkompatibel* ist! Betrachte folgendes Beispiel:

**Beispiel 8.1.** *GSP ist nicht anreizkompatibel*

Wir haben 3 Spieler und 3 Slots mit den privaten Bewertungen:  $v_1 = 10$ ,  $v_2 = 4$ ,  $v_3 = 2$ . Die Clickraten betragen:  $\alpha_1 = 200$ ,  $\alpha_2 = 199$ ,  $\alpha_3 = 0$ . Wenn alle Spieler ehrlich sind  $b_i = v_i$ , dann bekommt Spieler 1 den ersten Slot, zweithöchstes Gebot ist 4. Sein Nutzen, wenn er die Wahrheit sagt, entspricht:  $200 \cdot (10 - 4) = 1200$ . Wenn Spieler 1 nun  $b_i = 3$  sagt, dann bekommt er den zweiten Slot, dritthöchstes Gebot ist 2. Sein Nutzen, wenn er lügt, entspricht:  $199 \cdot (10 - 2) = 1592$ . ■

Wie können wir die resultierenden Zuweisungen und Anreize von GSP erfassen? Ergibt GSP gute Zuweisungen obwohl es mit unehrlichen Geboten arbeitet?

Jeder anreizkompatible oder nicht-anreizkompatible Mechanismus induziert ein strategisches Spiel unter den Spielern. Das Spiel für GSP ist gegeben wie folgt.

- $n$  Spieler, Strategieraum  $\Sigma_i = [0, \infty)$
- Spieler  $i$  wählt Strategie  $b_i \in \Sigma_i$ , Zustand  $b$  des Spiels
- Nummeriere Spieler  $v_1 \geq \dots \geq v_n$  bzgl. Wert (nicht Gebot)
- Sei  $\pi$  Ordnung bzgl. Gebot.  $\pi(j) = i$ , wenn  $i$  das  $j$ -größte Gebot hat.
- Nutzen von Spieler  $i$ :

$$u_i(b) = \alpha_j(v_i - b_{\pi(j+1)}) \quad \text{wobei } j \text{ mit } \pi(j) = i.$$

Ein Mechanismus ist anreizkompatibel genau dann, wenn  $b_i = v_i$  eine schwach dominante Strategie im Spiel ist. Wie oben gezeigt hat das GSP-Spiel diese Eigenschaft nicht immer. Anstatt IC betrachten wir hier (ex-post) Nash-Gleichgewichte des GSP-Spiels. Ein reines Nash-Gleichgewicht ist ein Zustand  $b$ , in dem kein Spieler unilateral zu einem anderen Gebot  $b'_i \neq b_i$  abweichen möchte (nachdem Auktion mit  $b$  stattgefunden hat).

Wir stellen zuerst fest, dass es sich nicht lohnt, ein Gebot  $b_i > v_i$  abzugeben.

**Lemma 8.2.** *Für jedes  $b_i \geq v_i$  und jedes  $b_{-i}$  gilt  $u_i(b_i, b_{-i}) \leq u_i(v_i, b_{-i})$ .*

*Beweis.* Wenn ein Spieler  $i$  in beiden Zuständen  $(b_i, b_{-i})$  und  $(v_i, b_{-i})$  den Slot  $j$  bekommt, erhält er in beiden Zuständen den gleichen Nutzen.

Ansonsten hebt ihn sein erhöhtes Gebot  $b_i > v_i$  auf Position  $k < j$  in  $(b_i, b_{-i})$ , und er schiebt Spieler  $\pi(k)$  runter zu Slot  $k + 1$ . Die neue Zahlung wird  $b_{\pi(k)} \geq v_i$  und sein Nutzen wird

$$\alpha_k(v_i - b_{\pi(k)}) \leq 0 \leq \alpha_j(v_i - b_{\pi(j+1)}) .$$

□

**Satz 8.3** (Edelman, Ostrovsky, Schwarz 2007). *Das GSP-Spiel hat ein reines Nash-Gleichgewicht mit maximalem sozialen Nutzen. Der Preis der Stabilität für reine Nash-Gleichgewichte ist 1.*

*Beweisidee.* Die Gebote im Nash-Gleichgewicht sind rekursiv gegeben durch

$$b_n = v_n \cdot \left(1 - \frac{\alpha_n}{\alpha_{n-1}}\right) \quad \text{und} \quad b_i = v_i - \frac{\alpha_i}{\alpha_{i-1}}(v_i - b_{i+1}) .$$

Wir zeigen nur, dass die resultierende Zuweisung den sozialen Nutzen maximiert, d.h. die Gebote sind geordnet wie Bewertungen:  $b_i \geq b_{i+1}$ ,  $\forall i = 1, \dots, n-1$ .

Dazu stellen wir erst fest, dass  $b_n \leq v_n$ . Wenn  $b_{i+1} \leq v_{i+1}$ , dann  $b_{i+1} \leq v_i$  und

$$b_i = v_i - \frac{\alpha_i}{\alpha_{i-1}}(v_i - b_{i+1}) \leq v_i .$$

Daher gilt mit Induktion, dass  $v_i \geq b_i$  für jeden  $i$ , und somit  $v_i \geq v_{i+1} \geq b_{i+1}$ . Das bedeutet

$$\begin{aligned} v_i - \frac{\alpha_i}{\alpha_{i-1}} \cdot v_i &\geq b_{i+1} - \frac{\alpha_i}{\alpha_{i-1}} \cdot b_{i+1} \\ \Rightarrow b_i = v_i - \frac{\alpha_i}{\alpha_{i-1}}(v_i - b_{i+1}) &\geq b_{i+1} \end{aligned}$$

und zeigt, dass die Zuweisung optimal ist. □

Wir wissen also, dass das beste reine Nash-Gleichgewicht optimal ist. Wenn die Spieler nicht überbieten, dann ist sogar das schlechteste reine Nash-Gleichgewicht nicht schlecht.

**Satz 8.4** (Paes Leme, Tardos 2010). *Im GSP-Spiel ist der Preis der Anarchie für reine Nash-Gleichgewichte ohne Überbieten höchstens 2.*

Den Satz werden wir später beweisen, dafür betrachten wir folgendes Lemma.

**Lemma 8.5.** *Für jeden Vektor von Bewertungen  $v$ , Klickraten  $\alpha$  und Ordnung  $\pi$  im Nash-Gleichgewicht, sowie für jedes  $i, j \in \{1, \dots, n\}$  gilt*

$$\frac{\alpha_j}{\alpha_i} + \frac{v_{\pi(i)}}{v_{\pi(j)}} \geq 1 . \quad (8.1)$$

*Damit gilt entweder  $\alpha_j/\alpha_i \geq 1/2$  oder  $v_{\pi(i)}/v_{\pi(j)} \geq 1/2$ .*

*Beweis.* Wenn  $j \leq i$ , dann gilt  $\alpha_j/\alpha_i \geq 1$ . Sonst, da  $b$  ein Nash-Gleichgewicht ist, will Bieter  $\pi(j)$  in Slot  $j$  nicht zu Slot  $i < j$  wechseln:

$$\alpha_j(v_{\pi(j)} - b_{\pi(j+1)}) \geq \alpha_i(v_{\pi(j)} - b_{\pi(i)}) .$$

Wir nutzen  $b_{\pi(j+1)} \geq 0$  und kein-Überbieten  $v_i \geq b_{\pi(i)}$ , um die linke Seite noch größer und die rechte Seite noch kleiner zu machen:

$$\alpha_j v_{\pi(j)} \geq \alpha_i(v_{\pi(j)} - v_{\pi(i)}) .$$

Teile beide Seiten durch  $\alpha_i v_{\pi(j)}$ , und das Lemma folgt. □

Das Lemma spielt eine entscheidende Rolle im Beweis des Satzes. Wir nennen eine Ordnung  $\pi$  schwach gültig, wenn damit die Bedingung (8.1) gilt. Das Lemma besagt, jede Ordnung  $\pi$  im Nash-Gleichgewicht ist schwach gültig.

*Beweis (Satz 8.4).* Wir zeigen die Schranke 2 für jeden Zustand, der eine schwach gültige Ordnung  $\pi$  erzeugt. Wir nutzen eine Induktion über die Anzahl der Spieler.

Der Induktionsanfang mit  $n = 1$  ist trivial und erfüllt.

Wir nehmen an der Preis der Anarchie ist 2, wenn  $\pi$  schwach gültig für  $n - 1$  Spieler.

Nun betrachten wir  $n$  Spieler. Sei  $\pi$  schwach gültig und sei  $j = \pi(1)$  der Spieler im ersten Slot und  $i = \pi^{-1}(1)$  der Slot des Spielers mit maximaler Bewertung. Wenn  $j = i = 1$ , lösche Spieler 1 und Slot 1. Die reduzierte Ordnung  $\pi$  bleibt schwach gültig für die verbleibenden  $n - 1$  Spieler und Slots. Mit der Induktionsannahme beschränken wir den sozialen Nutzen:

$$\alpha_1 v_1 + \sum_{k=2}^n \alpha_k v_{\pi(k)} \geq \alpha_1 v_1 + \frac{1}{2} \sum_{k=2}^n \alpha_k v_k \geq \frac{1}{2} \sum_{k=1}^n \alpha_k v_k .$$

Für allgemeines  $i$  und  $j$  nutzen wir deren Definition. Die Bedingung (8.1) sagt uns, dass entweder  $\alpha_i/\alpha_1 \geq 1/2$  oder  $v_j/v_1 \geq 1/2$ . Wir betrachten den ersten Fall ( $v_j/v_1 \geq 1/2$ ): Lösche Slot  $i$  und Spieler 1.  $\pi$  bleibt schwach gültig für die verbleibenden  $n - 1$  Spieler und Slots, also

$$\begin{aligned} \sum_{k \neq i} \alpha_k v_{\pi(k)} &\geq \frac{1}{2} (\alpha_1 v_2 + \dots + \alpha_{i-1} v_i + \alpha_{i+1} v_{i+1} + \dots + \alpha_n v_n) \\ &\geq \frac{1}{2} (\alpha_2 v_2 + \dots + \alpha_i v_i + \alpha_{i+1} v_{i+1} + \dots + \alpha_n v_n) \\ &= \frac{1}{2} \sum_{k=2}^n \alpha_k v_k \end{aligned}$$

und mit der Bedingung (erster Fall)

$$\sum_{k=1}^n \alpha_k v_{\pi(k)} = \alpha_i v_1 + \sum_{k \neq i} \alpha_k v_{\pi(k)} \geq \frac{1}{2} \alpha_1 v_1 + \frac{1}{2} \sum_{k=2}^n \alpha_k v_k .$$

Im zweiten Fall ( $\alpha_i/\alpha_1 \geq 1/2$ ) löschen wir Slot 1 und Spieler  $j$ .  $\pi$  bleibt schwach gültig für die  $n - 1$  verbleibenden Spieler und Slots, also

$$\begin{aligned} \sum_{k=2}^n \alpha_k v_{\pi(k)} &\geq \frac{1}{2} (\alpha_2 v_1 + \dots + \alpha_j v_{j-1} + \alpha_{j+1} v_{j+1} + \dots + \alpha_n v_n) \\ &\geq \frac{1}{2} (\alpha_2 v_2 + \dots + \alpha_j v_j + \alpha_{j+1} v_{j+1} + \dots + \alpha_n v_n) \\ &= \frac{1}{2} \sum_{k=2}^n \alpha_k v_k \end{aligned}$$

und mit der Bedingung (zweiter Fall)

$$\sum_{k=1}^n \alpha_k v_{\pi(k)} = \alpha_1 v_j + \sum_{k=2}^n \alpha_k v_{\pi(k)} \geq \frac{1}{2} \alpha_1 v_1 + \frac{1}{2} \sum_{k=2}^n \alpha_k v_k .$$

□ (Satz)

Betrachte ein gemischtes Nash-Gleichgewicht, in dem jeder Spieler eine *gemischte Strategie* wählt, d.h. eine Wahrscheinlichkeitsverteilung über  $\Sigma_i = [0, \infty)$ . Wir nehmen an, die Spieler *überbieten nicht*, d.h.  $\Pr[b_i \leq v_i] = 1$ .

Die Zuweisung, die sich aus  $\pi$  ergibt, ist nun eine Zufallsvariable. Wir nutzen  $\sigma = \pi^{-1}$ , also ist  $\sigma(i)$  die Position von Spieler  $i$  in  $\pi$ .

**Satz 8.6** (Paes Leme, Tardos 2010). *Im GSP-Spiel ist der Preis der Anarchie für gemischte Nash-Gleichgewichte ohne Überbieten höchstens 4.*

Der Beweis nutzt wieder ein technisches Lemma.

**Lemma 8.7.** *Wenn ein zufälliger Zustand  $b$  im GSP-Spiel verteilt ist gemäß eines gemischten Nash-Gleichgewichts ohne Überbieten, dann gilt*

$$\frac{\mathbb{E}[\alpha_{\sigma(i)}]}{\alpha_i} + \frac{\mathbb{E}[v_{\pi(i)}]}{v_i} \geq \frac{1}{2} .$$

*Beweis.* Wir zeigen zuerst folgende Behauptung:

Wenn  $i$  zu einer reinen Strategie  $b'_i = \min(v_i, 2 \cdot \mathbb{E}[b_{\pi(i)}])$  abweicht, dann bekommt er mit Wahrscheinlichkeit mind.  $1/2$  einen Slot in  $\{1, \dots, i\}$ .

Für den Beweis nehmen wir zuerst  $b'_i = v_i$  an. Dann gilt die Behauptung, denn ohne Überbieten haben maximal  $i - 1$  Spieler ein höheres Gebot. Nun sei  $b'_i = 2\mathbb{E}[b_{\pi(i)}]$ . Dann folgt mit der Markov-Ungleichung

$$\Pr[b_{\pi(i)} \geq b'_i] \leq \frac{\mathbb{E}[b_{\pi(i)}]}{b'_i} = \frac{1}{2} .$$

Das zeigt die Behauptung.

Damit ergibt sich

$$\begin{aligned} E[\alpha_{\sigma(i)}] \cdot v_i &\stackrel{\text{Zahlungen}}{\geq} E[u_i(b)] \\ &\stackrel{\text{Gem. NG}}{\geq} E[u_i(b'_i, b_{-i})] \\ &\stackrel{\text{Behauptung}}{\geq} \frac{1}{2} \alpha_i (v_i - b'_i) \\ &\stackrel{\text{min}=\text{max}}{\geq} \frac{1}{2} \alpha_i (v_i - 2E[b_{\pi(i)}]) \\ &\stackrel{\text{kein Überbieten}}{\geq} \frac{1}{2} \alpha_i (v_i - 2E[v_{\pi(i)}]) . \end{aligned}$$

Wenn wir beide Enden der Ungleichungskette durch  $\alpha_i \cdot v_i$  teilen, ergibt sich das Lemma.  $\square$

Der Satz kann nun direkt bewiesen werden.

*Beweis (Satz 8.1).* Dafür zeigen wir:

$$\begin{aligned} E \left[ \sum_{i=1}^n \alpha_{\sigma(i)} v_i \right] &\stackrel{\text{2 Arten erw. sozialen Nutzen zu schreiben}}{=} \frac{1}{2} E \left[ \sum_{i=1}^n \alpha_{\sigma(i)} v_i \right] + \frac{1}{2} E \left[ \sum_{i=1}^n \alpha_i v_{\pi(i)} \right] \\ &\stackrel{\text{Linearität des E.-Wertes}}{=} \frac{1}{2} \sum_{i=1}^n E[\alpha_{\sigma(i)}] \cdot v_i + \frac{1}{2} \sum_{i=1}^n \alpha_i \cdot E[v_{\pi(i)}] \\ &= \frac{1}{2} \sum_{i=1}^n \alpha_i v_i \left( \frac{E[\alpha_{\sigma(i)}]}{\alpha_i} + \frac{E[v_{\pi(i)}]}{v_i} \right) \\ &\stackrel{\text{Lemma}}{\geq} \frac{1}{4} \sum_{i=1}^n \alpha_i v_i . \end{aligned}$$

$\square$  (Satz)

Die momentan besten Schranken an den Preis der Anarchie finden sich bei [Caragiannis, Kalamanis, Kanellopoulos, Kyropoulou, Lucier, Paes Leme, Tardos 2012]:

- Reine NG:  $\leq 1.282$

- Reine NG:  $\geq 1.259$
- Gemischte NG, Grob-Korrelierte Ggw:  $\leq 2.310$
- Bayes'sche grob-korrelierte Ggw:  $\leq 2.927$  (Spiele mit unvollständiger Information)

Obwohl die Gebote manipuliert und nicht ehrlich sein können, liefert GSP im worst-case gute Zuweisungen, selbst für Stabilitätskonzepte mit relativ schwachen Anforderungen wie grob-korrelierte Gleichgewichte, und sogar ex-ante in Spielen mit unvollständiger Information.

In der Praxis werden auch Mechanismen genutzt, die nicht anreizkompatibel sind. Um die Güte von solchen Systemen zu beschreiben, betrachten wir Gleichgewichtskonzepte und den Preis der Anarchie.

## 8.2 Preis der Anarchie

Die *Erstpreisauktion* ist eine Ein-Gut-Auktion, in der der höchste Bieter das Gut erhält. Er zahlt sein Gebot, alle anderen Spieler zahlen nichts.

Wir definieren Erstpreisauktionen als Spiele:

- $n$  Spieler, Strategieraum  $\Sigma_i = [0, \infty)$ , Strategie  $b_i \in \Sigma_i$
- Ergebnismenge  $A = \{1, \dots, n\}$ , Zuweisungsfunktion  $f(b) = \arg \max_i b_i$
- Bewertung:  $v_i(i) = v_i \geq 0$  und  $v_i(j) = 0$  für  $j \neq i$
- Ordnung der Spieler:  $v_1 \geq v_2 \geq \dots \geq v_n$
- Zahlung:  $p_i(b) = b_i$  wenn  $f(b) = i$  und 0 sonst
- Nutzen:  $u_i(b) = v_i(f(b)) - p_i(b)$
- Spieler  $i$  wählt Gebot strategisch um seinen Nutzen zu maximieren.

**Proposition 8.8.** *Der Preis der Anarchie für reine Gleichgewichte in der Erstpreisauktion ist 1.*

*Beweis.* Sei  $b$  ein reines Nash-Gleichgewicht. Wir definieren eine Abweichung  $b'_1 = \sum_i p_i(b)$  für Spieler 1 und  $b'_i = 0$  für Spieler  $i \geq 2$ . Dann gilt

$$\begin{aligned} u_1(b) &\geq u_1(b'_1, b_{-1}) = v_1 - \sum_i p_i(b) = 0 \\ u_i(b) &\geq u_i(0, b_{-i}) = 0 \end{aligned} \quad \text{für alle } i \geq 2.$$

Sei  $\text{Opt}(v) = \max_a \sum_j v_j(a) = v_1$  der maximale soziale Nutzen, der erreicht wird, wenn der Spieler mit bestem Wert das Gut erhält. Damit gilt

$$\sum_i u_i(b'_i, b_{-i}) \geq \text{Opt}(v) - \sum_i p_i(b) . \quad (8.2)$$

Daraus folgt:

$$\begin{aligned} \sum_i v_i(f(b)) &= \sum_i u_i(b) + p_i(b) \\ &\geq \sum_i u_i(b'_i, b_{-i}) + p_i(b) \\ &\geq \text{Opt}(v) - \sum_i p_i(b) + \sum_i p_i(b) \\ &= \text{Opt}(v) . \end{aligned}$$

□

In der Erstpreisauktion existieren evtl. keine reinen Nash-Gleichgewichte. Wie können wir den Preis der Anarchie für allgemeinere Gleichgewichtskonzepte bestimmen, z.B. für gemischte oder korrelierte Gleichgewichte?

**Proposition 8.9.** *Der Preis der Anarchie für grob-korrelierte Gleichgewichte in der Erstpreisauktion beträgt maximal 2.*

*Beweis.* Sei  $\mathcal{V}$  ein grob-korreliertes Gleichgewicht, d.h.  $\mathcal{V}$  ist eine Verteilung über Zustände, so dass kein Spieler  $i$  zu einer einzigen Strategie  $b'_i$  abweichen will.

Wir betrachten so eine Abweichung:  $b'_1 = v_1/2$  und  $b'_i = 0$  sonst. Dann gilt natürlich  $\mathbb{E}_{b \sim \mathcal{V}}[u_i(b)] \geq \mathbb{E}_{b \sim \mathcal{V}}[u_i(0, b_{-i})] = 0$  für  $i \geq 2$ .

Wenn Spieler 1 das Gut nicht gewinnt, dann bietet und zahlt ein anderer Spieler  $j$  mehr als  $v_1/2$ , also  $u_1(b'_1, b_{-i}) = 0 \geq v_1/2 - p_j(b)$ . Wenn Spieler 1 das Gut gewinnt, dann  $u_1(b'_1, b_{-i}) = v_1/2$ . Damit gilt für jeden Zustand  $b$

$$u_1(b'_1, b_{-1}) \geq \frac{v_1}{2} - \sum_i p_i(b) .$$

Also:

$$\sum_i \mathbb{E}_{b \sim \mathcal{V}} [u_i(b'_i, b_{-i})] \geq \frac{1}{2} \text{Opt}(v) - \sum_i \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] \quad (8.3)$$

und somit

$$\begin{aligned} \sum_i \mathbb{E}_{b \sim \mathcal{V}} [v_i(f(b))] &= \sum_i \mathbb{E}_{b \sim \mathcal{V}} [u_i(b) + p_i(b)] \\ &\geq \frac{1}{2} \cdot \text{Opt}(v) - \sum_i \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] + \sum_i \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] \\ &= \frac{1}{2} \cdot \text{Opt}(v) . \end{aligned}$$

□

Im beiden Fällen haben wir die Gleichgewichtseigenschaft ausgenutzt. Wir haben eine Abweichung  $b'_i$  für jeden Spieler  $i$  bestimmt. Die Kernaussagen sind (8.2) und (8.3), daraus haben wir direkt einen Preis der Anarchie von 1 und 2 abgeleitet.

Im ersten Beweis hängt die Abweichung  $b'_1$  von  $b_{-1}$  ab. Damit klappt das Argument *nur für reine Nash-Gleichgewichte*, denn im gemischten Gleichgewicht kann ich meine Abweichung *nicht abhängig vom Ergebnis der Randomisierung der anderen Spieler* wählen.  $b'_1$  ist also so im gemischten Nash-Gleichgewicht nicht zulässig, daher bekommen wir dafür keine Schranke.

Im zweiten Beweis dagegen hängt keine der Abweichungen  $b'_i$  von  $b_i$  oder  $b_{-i}$  ab. Damit kann  $b'_i$  als ein mögliches "bestes Gebot im Nachhinein" angesehen werden und beschreibt somit eine mögliche Abweichung, die wir *sogar im grob-korrelierten Gleichgewicht* wählen könnten. Daher gilt die Schranke für grob-korrelierte Gleichgewichte.

Im allgemeinen Fall erlauben wir, dass die Abweichung  $b'_i$  von  $b_i$  abhängt, aber nicht von  $b_{-i}$ . Damit betrachten wir *korrelierte Gleichgewichte*. Die gleichen Argumente lassen sich (mit z.T. anderen Parametern und Schranken) für grob-korrelierte Gleichgewichte führen.

Welche Garantien können wir für Gleichgewichte aufstellen?

Betrachte einen beliebigen Mechanismus  $(f, p)$  mit *nicht-negativen Bewertungen*  $v \in V$ . Sei  $\text{Opt}(v) = \max_b \sum_i v_i(f(b))$  der optimale soziale Nutzen im Spiel.

**Definition 8.10** (Smoothness). Ein Mechanismus  $(f, p)$  mit Bewertungen aus  $V$  ist  $(\lambda, \mu)$ -smooth, wenn für jeden Zustand  $b$  ein Vektor von gemischten Strategien  $b'$  existiert, so dass

$$\sum_i \mathbb{E}_{b'_i} [u_i(b'_i, b_{-i})] \geq \lambda \cdot \text{Opt}(v) - \mu \cdot \sum_i p_i(b)$$

mit  $\lambda, \mu \geq 0$ . Strategie  $b'_i$  darf von  $v_i$  und  $b_i$  abhängen, aber *nicht* von den Strategien  $b_{-i}$  der anderen Spieler. Wir nennen  $b'_i$  die *Smoothness-Abweichung* von  $i$  in  $b$ .

Die Erstpreisauktion ist  $(1/2, 1)$ -smooth (siehe (8.3)). Das erlaubt eine Beschränkung des Preises der Anarchie – sofern die Spieler den *Mechanismus verlassen können* und dann Bewertung, Zahlung und Nutzen von 0 erhalten.

**Satz 8.11** (Syrgkanis, Tardos 2013). *Wenn ein Mechanismus  $(\lambda, \mu)$ -smooth ist und die Spieler den Mechanismus verlassen können, dann ist der Preis der Anarchie für korrelierte Gleichgewichte höchstens*

$$\frac{\max\{\mu, 1\}}{\lambda} .$$

*Beweis.* Sei  $\mathcal{V}$  das korrelierte Gleichgewicht. Dann gilt:

$$\begin{aligned} \sum_i \mathbb{E}_{b \sim \mathcal{V}} [v_i(f(b))] &= \sum_i \mathbb{E}_{b \sim \mathcal{V}} [u_i(b) + p_i(b)] \\ &\geq \sum_i \mathbb{E}_{b \sim \mathcal{V}} [u_i(b'_i, b_{-i}) + p_i(b)] \\ &\geq \lambda \cdot \text{Opt}(v) + (1 - \mu) \cdot \sum_i \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] \end{aligned}$$

Da ein Spieler den Mechanismus verlassen kann, gilt  $\mathbb{E}_{b \sim \mathcal{V}} [u_i(b)] \geq 0$ , und damit

$$\mathbb{E}_{b \sim \mathcal{V}} [v_i(f(b))] \geq \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] \geq 0 .$$

Somit

$$\begin{aligned} \lambda \cdot \text{Opt}(v) &\leq \sum_i \mathbb{E}_{b \sim \mathcal{V}} [v_i(f(b))] - (1 - \mu) \cdot \sum_i \mathbb{E}_{b \sim \mathcal{V}} [p_i(b)] \\ &\leq \max\{1, \mu\} \cdot \sum_i \mathbb{E}_{b \sim \mathcal{V}} [v_i(f(b))] . \end{aligned}$$

□

Wir betrachten kombinatorische Auktionen mit simultanen Gut-Geboten. Kombinatorische Auktionen seien gegeben als:

- $m$  Güter,  $n$  Spieler
- Spieler  $i$  hat Bewertungsfunktion  $v_i(S_i) \geq 0$  für  $S_i \subseteq [m]$

Wir betrachten simultane Ein-Gut-Auktionen als eine natürliche Variante für die Versteigerung von  $m$  Gütern.

- Jedes Gut wird mit einer separaten Ein-Gut-Auktion  $M_j$  versteigert
- Alle Ein-Gut-Auktionen  $M_j$  laufen simultan ab
- Spieler  $i$  gibt Gebot  $b_{ij} \geq 0$  in  $M_j$  ab,  $j = 1, \dots, m$
- $M_j$  weist Gut  $j$  zu (z.B. an höchsten Bieter  $\arg \max_i b_{ij}$ )
- $M_j$  bestimmt Zahlung  $p_{ij}(b_j)$  für jeden Spieler  $i$
- Spieler  $i$  erhält Güter  $S_i$  und Zahlungen  $\sum_j p_{ij}(b_j)$

Beachte, dass simultane Ein-Gut-Auktionen im Allgemeinen nicht anreizkompatibel sein können, da ein Spieler gar nicht seine Bewertung für Teilmengen von Gütern komplett mitteilen kann.

Wir beschränken uns auf *unit-demand* Bewertungen:

$$v_i(S_i) = \max_{j \in S_i} v_{ij}$$

mit  $v_{ij} \geq 0$  für alle  $j$ .

**Satz 8.12.** *Wenn jede Ein-Gut-Auktion  $(\lambda, \mu)$ -smooth ist und die Spieler unit-demand Bewertungen haben, dann ist auch die kombinatorische Auktion mit simultanen Gut-Geboten  $(\lambda, \mu)$ -smooth.*

*Beweis.* Betrachte  $S^*$ , die optimale Zuweisung der Güter. OBdA erhält jeder Spieler nur ein Gut in  $S^*$ . Sei  $S_i^* = \{j_i^*\}$  das *beabsichtigte Gut von  $i$* . Wir setzen nun  $v_i^*(S) = \max_i v_{ij}^*$  mit  $v_{ij}^* = v_{ij}$  wenn  $j = j_i^* \in S$  und 0 sonst.

Betrachte nun einen beliebigen Zustand  $b$  und die resultierende Allokation  $S$ . Dann gilt

$$\sum_i v_i(S_i) \geq \sum_i v_i^*(S_i) = \sum_{i: j_i^* \in S_i} v_{ij_i^*} ,$$

da wir in  $v_i^*$  nur den Wert des beabsichtigten Gutes erhalten, und auch nur wenn der Spieler das beabsichtigte Gut bekommt.

Wir konstruieren einen Vektor aus gemischten Strategien  $b'_i$  aus den Smoothness-Abweichungen  $b'_{ij}$  der Ein-Gut-Auktionen. Wir nehmen an, Spieler  $i$  hätte immer Wert  $v_{ij}^*$ , wenn er an Auktion  $M_j$  teilnimmt, für jedes  $j = 1, \dots, m$ . Er wählt die Smoothness-Abweichung  $b'_{ij}$ , die von  $v_{ij}^*$  und  $b_{ij}$  abhängt. Wenn  $b'_{ij}$  randomisiert ist, kombiniert er die Abweichungen in den Ein-Gut-Auktionen unabhängig zu einer Smoothness-Abweichung für alle Auktionen.

Sei nun  $X_{ij}$  das Event, dass Gut  $j$  an Spieler  $i$  gegeben wird, wenn  $i$  in  $M_j$  zu  $b'_{ij}$  abweicht. Dann gilt:

$$\mathbb{E}_{b'_i}[u_i(b'_i, b_{-i})] \geq \sum_j \mathbb{E}_{b'_i}[v_{ij}^* X_{ij} - p_{ij}(b'_{ij}, b_{-i,j})] .$$

Die Summe über Spieler und unabhängige Kombination der Abweichungen ergibt:

$$\sum_i \mathbb{E}_{b'_i}[u_i(b'_i, b_{-i})] \geq \sum_j \underbrace{\sum_i \mathbb{E}_{b'_{ij}}[v_{ij}^* X_{ij} - p_{ij}(b'_{ij}, b_{-i,j})]}_{(*)} .$$

Der markierte Term (\*) ist genau die Summe der erwarteten Nutzenfunktionen, wenn alle Spieler nur in  $M_j$  teilnehmen, Gebote  $b_{ij}$  vorliegen, jeder Spieler Wert  $v_{ij}^*$  für das Gut hat und einzeln zu  $b'_{ij}$  abweicht. Damit können wir nun die Smoothness-Eigenschaft für  $M_j$  anwenden.

Beachte, dass es für Gut  $j$  höchstens einen Spieler mit  $v_{ij}^* > 0$  gibt – den Spieler, der Gut  $j$  in  $S^*$  bekommt. Damit hat das Optimum für  $M_j$  mit Bewertungen  $v_{ij}^*$  den Wert  $\sum_i v_{ij}^*$ , und es ergibt sich:

$$\begin{aligned} \sum_i \mathbb{E}_{b'_i}[u_i(b'_i, b_{-i})] &\geq \sum_j \sum_i \mathbb{E}_{b'_{ij}}[v_{ij}^* X_{ij} - p_{ij}(b'_{ij}, b_{-i,j})] \\ &\geq \sum_j \left( \lambda \sum_i v_{ij}^* - \mu \sum_i p_{ij}(b_j) \right) \\ &= \lambda \sum_j \sum_i v_{ij}^* - \mu \sum_j \sum_i p_{ij}(b_j) \\ &= \lambda \sum_i v_i(S_i^*) - \mu \sum_i p_i(b) . \end{aligned}$$

□

Damit haben wir gezeigt, dass auch in kombinatorischen Auktionen mit mehreren Gütern und gleichzeitigen Erstpreisauktionen der Preis der Anarchie für korrelierte Gleichgewichte höchstens 2 ist. Durch die besondere Form der Abweichung (unabhängig von  $b_i$ ) ergibt sich die Schranke sogar für grob-korrelierte Gleichgewichte.

Diese Komposition der Smoothness-Eigenschaft gilt ebenso für die deutlich größere Klasse von *fraktional subadditiven* oder *XOS Bewertungen*. So eine Bewertungsfunktion  $v$  besteht aus einer Menge  $K$  von additiven Bewertungsfunktionen  $v_i^k(S) = \sum_{j \in S} v_{ij}^k$  für jedes  $k \in K$ . Die Bewertung einer Menge von Gütern ist dann die maximale Bewertung in einer der additiven Funktionen:  $v_i(S) = \max_{k \in K} v_i^k(S)$ .

XOS Bewertungen beinhalten eine ganze Reihe von interessanten und praktisch relevanten Fällen, z.B. additive, unit-demand, oder submodulare Bewertungen.

Die Analyse zeigt, dass in diesen Fällen simultane Erstpreisauktionen im Gleichgewicht sehr gute Zuweisungen erzeugen, obwohl sie nicht anreizkompatibel sind.

### 8.2.1 Zweitpreisauktionen

Die Zweitpreisauktion ist anreizkompatibel, d.h.  $b_i = v_i$  ist eine schwach dominante Strategie und somit auch ein reines Nash-Gleichgewicht (mit optimalem sozialen Nutzen). Es können aber durchaus noch weitere (schlechtere) Gleichgewichte existieren. Für die Analyse nutzen wir eine Formalisierung der Eigenschaft, dass die Spieler *nicht überbieten*.

**Definition 8.13** (Zahlungsbereitschaft). In Mechanismus  $(f, p)$  ist die maximale *Zahlungsbereitschaft* von Spieler  $i$  für Ergebnis  $a \in A$  bei Strategie  $b_i$  gegeben durch

$$z_i(b_i, a) = \max_{b_{-i}: f(b_i, b_{-i})=a} p_i(b_i, b_{-i}) .$$

**Definition 8.14** (ohne Überbieten). Eine Verteilung  $\mathcal{V}$  über Zustände *ohne Überbieten* erfüllt

$$\mathbb{E}_{b \sim \mathcal{V}}[z_i(b_i, f(b))] \leq \mathbb{E}_{b \sim \mathcal{V}}[v_i(f(b))] .$$

Die Definition von Smoothness benötigt in diesem Fall drei Parameter, wobei der dritte Parameter erst bei Mechanismen ohne Überbieten zu einer Schranke auf den Preis der Anarchie führt.

**Definition 8.15** (Schwache Smoothness). Ein Mechanismus  $(f, p)$  mit Bewertungen in  $V$  ist *schwach*  $(\lambda, \mu_1, \mu_2)$ -smooth, wenn für jeden Zustand  $b$  ein Vektor von gemischten Strategien  $b'$  existiert, so dass es  $\lambda, \mu_1, \mu_2 \geq 0$  gibt mit

$$\sum_i \mathbb{E}_{b'_i}[u_i(b'_i, b_{-i})] \geq \lambda \cdot \text{Opt}(v) - \mu_1 \cdot \sum_i p_i(b) - \mu_2 \sum_i z_i(b_i, f(b)) .$$

Beachte, dass jeder Mechanismus, der  $(\lambda, \mu)$ -smooth ist, auch schwach  $(\lambda, \mu, 0)$ -smooth ist – z.B. die Erstpreisauktion ist schwach  $(1/2, 1, 0)$ -smooth.

Die Zweitpreisauktion ist schwach  $(1, 0, 1)$ -smooth.

**Satz 8.16.** *Wenn ein Mechanismus schwach  $(\lambda, \mu_1, \mu_2)$ -smooth ist und die Spieler den Mechanismus verlassen können, dann ist der Preis der Anarchie für korrelierte Gleichgewichte ohne Überbieten höchstens*

$$\frac{\mu_2 + \max\{1, \mu_1\}}{\lambda} .$$

**Satz 8.17.** *Wenn jede Ein-Gut-Auktion schwach  $(\lambda, \mu_1, \mu_2)$ -smooth ist und die Spieler XOS Bewertungen haben, dann ist auch die kombinatorische Auktion mit simultanen Gut-Geboten schwach  $(\lambda, \mu_1, \mu_2)$ -smooth.*

Betrachte kombinatorische Auktionen, in denen jedes Gut in einer separaten Zweitpreisauktion verkauft wird (z.B. wenn alle Güter einzeln bei Ebay versteigert werden). Wenn die Bieter XOS-Bewertungen haben und ein korreliertes Gleichgewicht erreichen, dann ist der Preis der Anarchie höchstens 2.

### 8.3 Ansteigende Auktionen

Die Zweitpreisauktion ist eine Auktion mit *direkter Offenlegung*. Oft möchten Bieter ihr wahres Gebot für das Gut aber *lieber nicht offenlegen*, insbesondere wenn sie das Gut am Ende evtl. für einen deutlich kleineren Preis erhalten.

Das ist ein Grund dafür, dass *ansteigende Auktionen* in der Praxis verbreitet sind. Dabei wird der Preis so lange um eine Schrittweite  $\varepsilon$  erhöht bis nur noch ein Bieter im Rennen ist. Dieser Bieter erhält das Gut zum Preis, zu dem der vorletzte Bieter die Auktion verlassen hat.

Die ansteigende Auktion ist eine Zweitpreisauktion ohne direkte Offenlegung. Letztlich ergeben sich die gleichen Zuweisungen und (bei  $\varepsilon \rightarrow 0$ ) auch die gleichen Zahlungen wie in der Zweitpreisauktion.

Wir betrachten eine Variante für eine Menge  $[m]$  von  $m$  Gütern. Die Auktion nutzt *Bedarfsqueries* an ein *Bedarfsorakel* für jeden Bieter.

**Definition 8.18.** (Bedarfsorakel) Sei  $v_i : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$  die Bewertungsfunktion für Spieler  $i$ . Ein Bedarfsorakel beantwortet *Bedarfsqueries* der folgenden Form:

**Input:** Preis  $p_j \geq 0$  für jedes Gut  $j \in M$

**Output:** Eine Menge  $S \subseteq [m]$  von Gütern, die den Nutzen von  $i$  bei Preisen  $p_j$  maximiert.  
Formal: Ein *Bedarf* ist eine Menge  $S$ , so dass für alle  $S' \subseteq [m]$  gilt

$$v_i(S) - \sum_{j \in S} p_j \geq v_i(S') - \sum_{j \in S'} p_j .$$

Diese Arten von Orakelzugriff sind *effizient* und *vorteilhaft*. Spieler müssen sich *nicht immer über die genaue Bewertung aller Teilmengen* im Klaren sein.

---

#### Algorithmus 16: Ansteigende Auktion für mehrere Güter

---

**Input:** Menge  $[m]$  von Gütern, Menge  $\mathcal{N}$  der Spieler

**Output:** Zuweisung  $(S_1, \dots, S_n)$  der Güter an Spieler,  
ein Preis  $p_j \geq 0$  für jedes Gut  $j \in [m]$

/\* Initialisierung: \*/

Setze  $p_j \leftarrow 0, \forall j \in M$  und  $S_i \leftarrow 0, \forall i \in \mathcal{N}$

/\* Anstieg: \*/

Repeat:

    Für jedes  $i$ , sei  $D_i$  ein Bedarf von  $i$  bei Preisen  $p_j$  mit:

$p_j$  für  $j \in S_i$  und  $p_j + \epsilon$  für  $j \notin S_i$

**if**  $S_i = D_i$  **für jedes**  $i$  **then**

        └ exit loop

    Finde Spieler  $i$  mit  $S_i \neq D_i$  und aktualisiere:

        Für jedes  $j \in D_i \setminus S_i$  setze  $p_j \leftarrow p_j + \epsilon$

$S_i \leftarrow D_i$

        Für jedes  $k \neq i$  setze  $S_k \leftarrow S_k \setminus D_i$

/\* Ausgabe: \*/

return Zuweisung  $S_1, \dots, S_n$  und Preise  $p_1, \dots, p_m$

---

Wir nehmen an, dass die Bewertung  $v_i(S) \geq 0$  eine ganze Zahl ist, für jeden Spieler  $i$  und jede Menge  $S \subseteq [m]$ .

**Definition 8.19** (Substitutionsbewertung). Seien  $p$  und  $q$  zwei Preisvektoren mit  $p_j \geq q_j$  für alle  $j \in [m]$ . Bewertung  $v_i$  ist eine *Substitutionsbewertung*, wenn jeder Bedarf bei Preisen  $p$  alle Güter des Bedarfs bei  $q$  enthält, deren Preis gleich geblieben ist.

Formaler: Für jedes  $S^q \in \arg \max\{v_i(S) - \sum_{j \in S} q_j\}$  gibt es ein  $S^p \in \arg \max\{v_i(S) - \sum_{j \in S} p_j\}$  mit  $S^p \supseteq \{j \in S^q \mid p_j = q_j\}$ .

**Beispiel 8.20.** *Substitutionsbewertungen*

$$\begin{array}{ll} \text{Additiv} & v_i(S) = \sum_{j \in S} v_{ij} \\ \text{Unit-Demand} & v_i(S) = \max_{j \in S} v_{ij} \\ \text{Konkav Multi-Unit} & v_i(S) = \sum_{j=1, \dots, |S|} v_j \text{ mit } v_1 \geq v_2 \geq \dots \end{array}$$

**Definition 8.21** (Walras-Gleichgewicht). Ein *Walras-Gleichgewicht* besteht aus einer Zuweisung  $(S_1, \dots, S_n)$  von Gütern und einem Vektor  $p = (p_1, \dots, p_m)$  von nicht-negativen Preisen, so dass  $S_i$  ein Bedarf für Spieler  $i$  ist bzgl. der Preise  $p$ . Wenn ein Gut  $j$  nicht zugewiesen wird, dann hat es Preis  $p_j = 0$ .

Ein Walras-Gleichgewicht hat einen Preis für jedes Gut. Jeder Spieler bekommt eine Menge von Gütern, die er bei den momentanen Preisen am liebsten haben möchte. Güter, die niemand haben will, erhalten Preis 0.

**Satz 8.22.** Sei  $v_i$  eine ganzzahlige Substitutionsbewertung und  $\varepsilon < 1/m$ . Dann ist die Ausgabe der ansteigenden Auktion ein Walras-Gleichgewicht. Die Zuweisung maximiert den sozialen Nutzen  $\sum_i v_i(S_i)$ .

*Beweis.* Wir zeigen zuerst eine *Behauptung*: Zu jeder Zeit gilt für jeden Spieler  $S_i \subseteq D_i$ .

Die Behauptung gilt am Anfang, betrachte ein Update für Spieler  $i$ . Für  $i$  selbst gilt  $S_i = D_i$  am Ende des Updates. Für  $k \neq i$  können zwei Änderungen auftreten:

1. Die Güter wurden von  $i$  aus  $S_k$  weggenommen. Dann wird  $S_k$  kleiner ohne Einfluß auf  $D_k$ . Die Behauptung gilt weiterhin.
2. Die Preise der Güter außerhalb von wurden  $S_k$  erhöht, aber das betrifft  $S_k$  nicht. Da  $v_i$  eine Substitutionsbewertung ist, gilt: Die Güter, die aus  $D_k$  entfernt werden, haben einen erhöhten Preis, aber sie liegen nicht in  $S_k$ . Die Behauptung gilt weiterhin.

Güter werden also nur von anderen Spielern *gestohlen*, aber kein Gut wird einfach *weggeworfen*. Wenn also ein Gut zum ersten Mal zugewiesen wird und sich sein Preis auf  $> 0$  erhöht, dann bleibt es für immer zugewiesen. Es gibt es kein Gut  $j$ ,

- das zugewiesen ist und Preis  $p_j = 0$  hat, oder
- das nicht zugewiesen ist und Preis  $p_j > 0$  hat.

Der Beweis zur Optimalität der Zuweisung ist eine Übungsaufgabe. □

In dem Beweis sind wir von Substitutionsbewertungen ausgegangen. Für allgemeine Bewertungen kann das *Exposure Problem* auftreten.

Bei einer Auktion mit z.B. Schuhen hat Alice eine Komplementärbewertung. Sie bietet auf beide Schuhe eines Paares, die sie beide positiv bewertet. Die Auktion könnte einen Schuh wegnehmen und anderem Bieter geben. Dann Alice verlässt die Auktion mit Wert 0.

Bei Auktionen mit Substitutionsbewertungen *reduzieren* Bieter ihren *Bedarf*, um den eigenen Nutzen zu erhöhen.

Walras-Gleichgewichte sind anfällig für preisantizipierende Bieter. Wenn ein Bieter eine kleinere Bedarfsmenge mitteilt, bleiben die Preise deutlich kleiner, und der Nutzen ist höher als beim ehrlichem Gebot.

Bieter können ihre Gebote zum *Signalisieren in Auktionen* verwenden. Dies geschah z.B. bei einer Spektrumsauktion für Mobilfunklizenzen in den USA, bei der die Kommunikationsfirmen USWest und McLeod in Rochester die Lizenz #378 kaufen wollten. Dafür boten sie auf präferierte Lizenzen ihrer Gegner in anderen Gebieten und wählten die Ziffern 378 als letzte Stellen ihrer Gebote.

### 8.3.1 Spektrumsauktionen

Im Gegensatz zur Sponsored Search, wo Auktionen täglich millionenfach ablaufen, werden Auktionen für Spektren in Funknetzwerken nur sehr selten und in sehr großem Umfang durchgeführt. Hierbei wird eine Menge von Frequenzspektren (z.T. gebündelt in Kanälen) vom Staat an Telekommunikationsanbieter versteigert. Die letzten Auktionen in Deutschland waren die Versteigerungen der Lizenzen zu Frequenzen im Bereich 700-1800 MHz im August 2015<sup>1</sup>.

Formate wie sequentielle Ein-Kanal-Auktionen (Schweiz, 2000) oder Auktionen mit versiegelttem Gebot (Neuseeland, 1990) haben zu teilweise skurrilen Ergebnissen und Preisen geführt. Dagegen haben sich simultane ansteigende Auktionen hierbei bewährt.

In den USA wurde 2016 ein interessanter neuer Ansatz mit *Rückwärtsauktionen* genutzt. Dabei wurden Frequenzbänder von Fernsehsendern zurückgekauft und dann neu gepackt an die Sender vergeben. Das erlaubt z.B. die zusammenhängende Zuweisung von Frequenzbändern und macht die resultierende Zuweisung deutlich kompakter. Das frei werdende Spektrum kann dann weiter verkauft werden z.B. an Mobilfunkanbieter.

Dieser Ansatz kann vereinfacht durch ein Single-Parameter Modell beschrieben werden:

- $n$  TV Sender, Sender  $i$  hat private Bewertung  $v_i \geq 0$  für seinen Kanal
- Kanal wird nicht zurückgekauft, Nutzen  $u_i = 0$ . ( $i$  verliert),
- Kanal wird zurückgekauft zum Preis  $p$ , Nutzen  $p - v_i$ . ( $i$  gewinnt)

Für eine *gültige Menge* von Siegern  $W \subseteq \mathcal{N}$  können die Kanäle zusammengelegt und neu gepackt werden. Ob eine Menge  $W$  gültig ist, wird durch ein Graphenfärbungsproblem entschieden, da zwei TV-Sender im gleichen Einzugsgebiet aus Interferenzgründen nicht den gleichen Kanal bekommen können. Obwohl dieses Gültigkeitsproblem i.A. NP-hart ist, wird es hier für mittelgroße Instanzen mit Ansätzen aus dem SAT-Lösen exakt gelöst.

Wie sollte die gültige Menge an Siegern in der Rückwärtsauktion gewählt werden? Für dieses Problem wurde ein Greedy Mechanismus benutzt, der sehr ähnlich ist zu dem Mechanismus, den wir für Rucksackauktionen betrachtet haben. Diesmal läuft er allerdings umgekehrt ab – Bieter werden einzeln aus der Siegermenge entfernt.

---

#### Algorithmus 17: Deferred Allocation

---

```

Setze  $W = \mathcal{N}$ 
while  $\exists i \in W$  mit  $W \setminus \{i\}$  gültig do
   $\perp$  Entferne ein solches  $i$  aus  $W$ 
return  $W$ 

```

---

Es wird ein Bieter entfernt, solange die resultierende Menge noch gültig ist. Für die Wahl des Bieters kann man eine *Scoring-Funktion* benutzen (z.B. höchstes Gebot, höchstes Verhältnis zur Größe des Marktes). Wenn die Scoring-Funktion monoton im Gebot des Bieters und unabhängig von den anderen Geboten ist, dann wird der Mechanismus anreizkompatibel.

---

<sup>1</sup>siehe auch [https://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen\\_Institutionen/Frequenzen/Projekt2016\\_Frequenzauktion/projekt2016-node.html](https://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Projekt2016_Frequenzauktion/projekt2016-node.html)

## 8.4 Zusammenfassung

In vielen Begebenheiten werden Mechanismen, die nicht anreizkompatibel sind, verwendet, das liegt zum Beispiel daran, dass die Bieter bei gleichen Geboten höher Zahlungen leisten (vergleiche VCG und GSP). Obwohl die Bieter bei solchen Mechanismen nicht mit Gewissheit ihren wahren Wert nennen, besitzt GSP konstante PoA und liefert daher gute Ergebnisse. Diese Beobachtung gilt auch für Erst-Preis-Auktionen mit einem Gut und kombinatorische Auktionen bestehend aus separaten Zweitpreisauktionen.

Die ansteigende Auktion wird in der Praxis häufig angewandt und hat den Vorteil, dass die Bieter kein genaues Gebot abgeben müssen. Es reicht jeweils zu entscheiden, ob sie weiterhin mitbieten. Wenn die Bieter Substitutionsbewertungen verwenden, dann ist die Ausgabe der Auktion ein Walras-Gleichgewicht und maximiert den sozialen Nutzen. Allerdings ist das Auktionsmodell problemfälliger z.B. für preisantizipierende Bieter, wodurch möglicherweise nur geringe Preise gezahlt werden.

Zur Vertiefung wird folgende Literatur empfohlen:

- Edelman, Ostrovsky, Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review* 97(1):242-259, 2007.
- Caragiannis, Kaklamanis, Kanellopoulos, Kyropoulou, Lucier, Paes Leme, Tardos. Bounding the Inefficiency of Outcomes in Generalized Second Price Auctions. *Journal of Economic Theory* 156:343-388, 2015.
- Syrgkanis, Tardos. Composable and Efficient Mechanisms. *STOC* 2013
- Nisan, Roughgarden, Tardos, Vazirani. *Algorithmic Game Theory*, 2007. (Kapitel 11)
- Roughgarden. *Twenty Lectures on Algorithmic Game Theory*, 2016. (Kapitel 8)

# 9 | Mechanismen ohne Geld

Bisher haben wir Geld verwendet, um durch die Zahlungen Anreize für die Spieler zu schaffen. So konnten wir anreizkompatible Mechanismen und Spiele mit guten Gleichgewichten entwerfen. Wahlen stellen ein natürliches Beispiel für Mechanismen ohne Geld dar. Wie können entsprechende Mechanismen entworfen werden, die auch ohne Geld gute Ergebnisse erzielen?

## 9.1 Wahlen

Wir definieren Wahlen durch:

- Menge  $A$  von *Kandidaten* (oder Ergebnissen, Alternativen)
- Menge  $\mathcal{N}$  von  $n$  *Wählern* (oder Spielern)
- Menge  $L$  der möglichen *Präferenzen* (totale Ordnungen von  $A$ )
- Jeder Wähler  $i$  hat eine *Präferenz* (oder Präferenzordnung)  $\succ_i \in L$  über die Kandidaten  $A$
- Eine *soziale Nutzenfunktion* ist eine Funktion  $F : L^n \rightarrow L$ .
- Eine *soziale Auswahlfunktion* ist eine Funktion  $f : L^n \rightarrow A$ .

Eine soziale Auswahlfunktion gibt nur einen einzigen Sieger aus, eine soziale Nutzenfunktion gibt eine vollständige Rangliste aller Kandidaten aus.

Wir betrachten zuerst zwei intuitive soziale Nutzenfunktionen.

### Beispiel 9.1. Mehrheitswahl und Condorcet-Paradox

| Wähler | Präferenzordnung |
|--------|------------------|
| 1      | M S              |
| 2      | S M              |
| 3      | M S              |

Wir führen eine direkte Kanzlerwahl über die Kandidaten (M) und (S) durch. Das Resultat wird per Mehrheitswahl bestimmt, wobei wir die Kandidaten paarweise betrachten und gegeneinander antreten lassen. Der Kandidat, der in der Mehrheit gewinnt, steht in der Präferenzliste weiter vorne. In diesem Fall resultieren die genannten Präferenzordnungen in M,S.

| Wähler | Präferenzordnung |
|--------|------------------|
| 1      | M S              |
| 2      | M S              |
| 3      | M S              |

Für zwei Kandidaten setzt die Regel viele gute Eigenschaften um, so wird die Mehrheit der Präferenzen repräsentiert und jeder Kandidat befindet sich in der Position, in der er/sie am meisten genannt wird. Des Weiteren ist strategisches Wählen *nicht profitabel*: Angenommen ein Wähler mit Präferenz wie die Mehrheit ändert sein Votum. Dann verändert sich das Resultat nicht zu seinen Gunsten. Ein Wähler mit Präferenz entgegen

der Mehrheit kann Ergebnis nicht ändern.

Diese Eigenschaften gelten nicht mehr, wenn die Mehrheitswahl auf mehr als zwei Kandidaten angewandt wird. Wir betrachten eine direkte Kanzlerwahl über die Kandidaten (M), (S) und (N).

Wenn paarweise alle Kandidaten verglichen werden, ergibt die Mehrheitswahl einen Kreis: Jeweils 2 Wähler mögen M lieber als S, S lieber als N, und N lieber als M ... Diese Instanz zeigt, dass die kollektive Präferenz widersprüchlich sein kann (kreisend, nicht transitiv), obwohl jede individuelle Präferenz wohl-definiert ist. Das Beispiel wird *Condorcet-Paradox* genannt und wurde vom Marquis de Condorcet um 1785 herum entdeckt.

| Wähler | Präferenzordnung |
|--------|------------------|
| 1      | M S N            |
| 2      | S N M            |
| 3      | N M S            |

■

**Beispiel 9.2. Pluralitätswahl**

| Wähler | Präferenzordnung |
|--------|------------------|
| 1      | M S N            |
| 2      | S N M            |
| 3      | N M S            |

| Wähler | Genannte Ordnung |
|--------|------------------|
| 1      | M S N            |
| 2      | S N M            |
| 3      | N S M            |

Wir betrachten die Pluralitätsregel (plurality rule), bei der jeder Kandidat auf die Position in der Rangliste kommt, auf der er am häufigsten bei den Wählern auftritt. Wir lösen Ties bzgl. alphabetischer Ordnung. Mit dieser Regel folgt im Beispiel das Resultat M, N, S. Hier ist strategisches Wählen profitabel für den dritten Wähler! Wenn dieser seine genannte Ordnung teilweise verändert, entspricht das Resultat N, S, M, wodurch sich das Ergebnis für ihn verbessert. Wie können wir strategisches Wählen vermeiden? Trivialerweise können wir einen Wähler als Diktator bestimmen, der das Ergebnis diktiert durch seine genannte Präferenz. Aber bekommen wir das auch anders hin?

■

**9.1.1 Unmöglichkeitsergebnisse**

Für weitere Überlegungen brauchen wir die konkreten Eigenschaften für gültige soziale Nutzenfunktionen:

- *Einstimmigkeit*: Für jedes  $\succ \in L$  gilt  $F(\succ, \dots, \succ) = \succ$ .
- Wähler  $i$  ist ein *Diktator* in einer sozialen Nutzenfunktion wenn für alle  $\succ_1, \dots, \succ_n \in L$  gilt  $F(\succ_1, \dots, \succ_n) = \succ_i$ . Dann wird  $F$  eine *Diktatur* genannt.
- *Unabhängigkeit von irrelevanten Alternativen (IIA)*: Die soziale Präferenz zwischen je zwei Kandidaten  $a$  und  $b$  hängt nur von den Präferenzen der Wähler bzgl.  $a$  und  $b$  ab. Formal: Für jede  $a, b \in A$  und jede  $\succ_1, \dots, \succ_n, \succ'_1, \dots, \succ'_n \in L$ , sei  $\succ = F(\succ_1, \dots, \succ_n)$  und  $\succ' = F(\succ'_1, \dots, \succ'_n)$ . Wenn  $a \succ_i b \Leftrightarrow a \succ'_i b$  für alle  $i$ , dann gilt  $a \succ b \Leftrightarrow a \succ' b$ .

Im Beispiel 9.2 der Pluralitätswahl wird IIA verletzt. Durch das Abweichen von Spieler 3 verändert sich die Ordnung des Paares (M,N), obwohl jeder Wähler bzgl. M und N die gleiche paarweise Präferenz hat in beiden Instanzen.

**Lemma 9.3** (Pairweise Neutralität). *Seien  $\succ_1, \dots, \succ_n$  und  $\succ'_1, \dots, \succ'_n$  zwei Präferenzprofile, und  $\succ = F(\succ_1, \dots, \succ_n)$  und  $\succ' = F(\succ'_1, \dots, \succ'_n)$ . Wenn für jeden Wähler  $i$  gilt  $a \succ_i b \Leftrightarrow c \succ'_i d$ , dann  $a \succ b \Leftrightarrow c \succ' d$ .*

*Beweis.* Wir benennen zuerst die Kandidaten um, so dass  $a \succ b$  und  $c \neq b$  (aber evtl.  $a = c$ , und/oder  $b = d$ ). Jetzt passen wir  $\succ_i$  und  $\succ'_i$  an, damit sie identisch werden bzgl.  $a, b, c, d$ . Wir bewegen  $c$  und  $d$  in  $\succ_i$ , sowie  $a$  und  $b$  in  $\succ'_i$ , so dass sich  $c$  vor  $a$  und  $b$  vor  $d$  befindet:

$$\begin{array}{lcl} \succ_1: & \dots, a, \dots, b, \dots & \rightarrow \dots, c, a, \dots, b, d, \dots \\ \succ'_1: & c, \dots, d, \dots & \rightarrow c, a, \dots, b, d, \dots \end{array}$$

$$\begin{aligned} \succ_2: & \quad \dots, b, \dots, a, \dots & \rightarrow & \quad \dots, b, d, \dots, c, a, \dots \\ \succ'_2: & \quad \dots, d, c, \dots & \rightarrow & \quad \dots, b, d, c, a, \dots \end{aligned}$$

und so weiter...

Per Annahme gilt  $a \succ_i b \Leftrightarrow c \succ'_i d$ . IIA garantiert, dass  $a$  und  $b$  in der gleichen Ordnung bleiben in  $\succ$ ;  $c$  und  $d$  bleiben in der gleichen Ordnung in  $\succ'$ . Ebenfalls mit IIA folgt, dass wir nun alle anderen Elemente beliebig umstellen können. Dann gilt  $\succ'_i = \succ_i$ . Mit Einstimmigkeit gilt nun  $c \succ a$  und  $b \succ d$ , also  $c \succ d$ . Mit  $\succ_i = \succ'_i$  für alle  $i$  erhalten wir auch  $c \succ' d$ .  $\square$

**Satz 9.4** (Arrow, 1950). *Jede soziale Nutzenfunktion über einer Menge  $|A| \geq 3$  Kandidaten, die Einstimmigkeit und IIA erfüllt, ist eine Diktatur.*

Sei  $F$  eine soziale Nutzenfunktion mit den genannten Eigenschaften (Einstimmigkeit und IIA).

*Beweis.* Paarweise Neutralität zeigt, dass eine soziale Nutzenfunktion, die Einstimmigkeit und IIA erfüllt, einen allgemein zugrunde liegenden Ansatz verfolgt, die globale Präferenz zu bestimmen. Dieser Ansatz ist ähnlich für alle genannten Ordnungen und alle paarweisen Vergleiche zwischen Kandidaten. Diese Einsicht kann man für den Beweis nutzen, dass die Funktion sogar eine Diktatur ist.

Sei  $a \neq b$  und  $c \neq d$ . Wenn es keine Wähler mit  $a \succ_i b$  gibt, dann ist  $b \succ a$ . Analog gilt, wenn es nur Wähler mit  $a \succ_i b$  gibt, dann ist  $a \succ b$ . Zuerst geben alle Wähler  $b \succ_i a$  an. Dann lassen wir schrittweise einen weiteren Wähler zu  $a \succ_i b$  abweichen. OBdA tritt der erste Wechsel des Resultats bei Wähler  $i^*$  auf:

|               |     |           |               |     |   |             |
|---------------|-----|-----------|---------------|-----|---|-------------|
| 1             | ... | $i^* - 1$ | $i^*$         | ... | n | Resultat    |
| $a \succ_i b$ |     |           | $b \succ_i a$ |     |   | $b \succ a$ |
| $a \succ_i b$ |     |           | $b \succ_i a$ |     |   | $a \succ b$ |

Behauptung:  $i^*$  ist der Diktator!

$i^*$  ist ein Diktator, wenn  $c \succ_{i^*} d \Rightarrow c \succ d$  für alle  $c \neq d \in A$ . Betrachte eine beliebige Menge an Präferenzen mit  $c \succ_{i^*} d$  und ein drittes Element  $e \in A$  mit  $e \neq c$  und  $e \neq d$ . Verschiebe  $e$ , so dass sich  $\succ_i$  schematisch ergibt wie folgt:

|       |     |     |     |     |     |     |     |
|-------|-----|-----|-----|-----|-----|-----|-----|
| 1     | $e$ | ... |     |     |     |     |     |
| ...   | $e$ | ... |     |     |     |     |     |
| $i^*$ | ... | $c$ | ... | $e$ | ... | $d$ | ... |
| ...   | ... |     |     |     |     | $e$ |     |
| $n$   | ... |     |     |     |     | $e$ |     |

Wegen IIA ändert das nicht die Ordnung von  $c$  und  $d$  in  $\succ$ .  $(e, c)$  steht in den Ordnungen nun genau wie vorher  $(a, b)$ . Mit paarweiser Neutralität gilt  $c \succ e$  und mit gleichem Argument sehen wir:  $e \succ d$ . Damit gilt also  $c \succ d$ . Da  $c$  und  $d$  beliebige Elemente waren, bestimmt die Präferenz von  $i^*$  alle Paare und damit das ganze Ergebnis.  $\square$

Für gültige soziale Auswahlfunktion können die folgenden Eigenschaften gelten:

- $f$  kann *strategisch manipuliert* werden durch Wähler  $i$ , wenn es  $\succ_1, \dots, \succ_n$  und  $\succ'_i$  gibt mit  $a \succ_i b$ , wobei  $b = f(\succ_1, \dots, \succ_n)$  und  $a = f(\succ_1, \dots, \succ'_i, \dots, \succ_n)$ .  $f$  wird *anreizkompatibel* (*incentive compatible, IC*) genannt, wenn es nicht strategisch manipuliert werden kann.
- $f$  ist *monoton*, wenn aus  $f(\succ_1, \dots, \succ_n) = a \neq b = f(\succ_1, \dots, \succ'_i, \dots, \succ_n)$  folgt, dass  $a \succ_i b$  und  $b \succ'_i a$ .
- Wähler  $i$  ist ein *Diktator* in  $f$ : Für jede  $\succ_1, \dots, \succ_n \in L$  gilt, dass wenn  $a \succ_i b$  für alle  $b \neq a$ , dann  $f(\succ_1, \dots, \succ_n) = a$ . Dann ist  $f$  eine *Diktatur*.

- $f$  ist *surjektiv* oder *ausschöpfend* in  $A$ , wenn es für jeden Kandidaten  $a \in A$  eine Menge von Präferenzen gibt, bei der  $a$  der Sieger ist.

**Proposition 9.5.** *Eine soziale Auswahlfunktion  $f$  ist anreizkompatibel genau dann, wenn  $f$  monoton ist.*

*Beweis.* Die Proposition ist eine direkte Folgerung aus der Definition. □

**Satz 9.6** (Gibbard 1973; Satterthwaite 1975). *Sei  $f$  eine soziale Auswahlfunktion  $f$ , die surjektiv ist mit  $|A| \geq 3$ .  $f$  ist anreizkompatibel genau dann, wenn  $f$  eine Diktatur ist.*

*Beweis.* Wir beweisen die nicht-triviale Richtung des Satzes. Für jede monotone soziale Auswahlfunktion  $f$ , die ausschöpfend in  $A$  ist, leiten wir eine soziale Nutzenfunktion  $F$  her, die Einstimmigkeit und IIA erfüllt.

Sei  $\succ$  eine Präferenzordnung und  $S \subset A$ . Wir nennen  $\succ^S$  die Ordnung, in der alle Elemente von  $S$  in ihrer Ordnung in  $\succ$  nach vorn geschoben werden.

Beispiel:  $S = \{a, b, c\}$ ,  $A = S \cup \{d, e, f\}$

|     |         |     |     |     |     |               |           |     |     |     |     |
|-----|---------|-----|-----|-----|-----|---------------|-----------|-----|-----|-----|-----|
|     | $\succ$ |     |     |     |     | $\rightarrow$ | $\succ^S$ |     |     |     |     |
| $a$ | $e$     | $d$ | $c$ | $f$ | $b$ | $a$           | $c$       | $b$ | $e$ | $d$ | $f$ |
| $b$ | $f$     | $e$ | $d$ | $a$ | $c$ | $b$           | $a$       | $c$ | $f$ | $e$ | $d$ |

Sei  $F$  eine  $f$ -erweiternde soziale Nutzenfunktion mit  $F(\succ_1, \dots, \succ_n) = \succ$ , wobei  $a \succ b$  genau dann, wenn  $f(\succ_1^{\{a,b\}}, \dots, \succ_n^{\{a,b\}}) = a$

Es ist erstmal gar nicht klar, ob  $F$  auf diese Weise überhaupt eine konsistente Ordnung  $\succ$  erzeugt, also eine gültige soziale Nutzenfunktion darstellt. Dafür verwenden wir folgendes Lemma.

**Lemma 9.7.** *Wenn  $f$  eine surjektive, anreizkompatible soziale Auswahlfunktion ist, dann ist die Erweiterung  $F$  eine soziale Nutzenfunktion.*

Um zu beweisen, dass  $F$  eine konsistente soziale Nutzenfunktion darstellt, müssen wir Antisymmetrie und Transitivität zeigen.

**Lemma 9.8.** *Wenn  $f$  eine surjektive anreizkompatible soziale Auswahlfunktion und keine Diktatur ist, dann erfüllt die Erweiterung  $F$  Einstimmigkeit, IIA und ist auch keine Diktatur.*

Das ergibt einen Widerspruch mit dem Satz von Arrow.

Für die Beweise beider Lemmata benötigen wir die folgende Behauptung.

**Behauptung 9.9.** *Für jedes  $\succ_1, \dots, \succ_n$  und jedes  $S$  ist der Sieger  $f(\succ_1^S, \dots, \succ_n^S) \in S$ .*

*Beweis.*  $f$  ist ausschöpfend, also gibt es  $\succ_1'', \dots, \succ_n''$  mit einem Sieger  $a \in S$ .

Transformiere  $\succ_1'', \dots, \succ_n''$  in  $\succ_1^S, \dots, \succ_n^S$ :

- 1: Starte mit  $\succ_1'', \dots, \succ_n''$
- 2: Bewege nacheinander die Elemente von  $S$  nach vorne
- 3: Sortiere die Elemente in  $A \setminus S$  hinten um
- 4: Sortiere die Elemente in  $S$  vorne um

$\Rightarrow \succ_1^S, \dots, \succ_n^S$  erreicht.

Monotonie stellt sicher, dass kein  $b \notin S$  jemals zum Sieger wird während dieser Transformation.

Also  $f(\succ_1^S, \dots, \succ_n^S) \in S$ .

□ [Behauptung]

*Beweis (Lemma 9.7).*

- **Antisymmetrie:** Wenn  $a \succ b$  und  $b \succ a$ , dann  $a = b$ .  
Wenn  $a \succ b$  und  $b \succ a$ , dann  $a = b$ . Gilt, da  $f(\succ_1^{\{a,b\}}, \dots, \succ_n^{\{a,b\}}) \in \{a, b\}$ .

- Transitivität: Wenn  $a \succ b$  und  $b \succ c$ , dann  $a \succ c$ .  
 Wenn  $a \succ b$  und  $b \succ c$ , dann  $a \succ c$ . Für einen Widerspruch nehmen wir an, dass  $a \succ b \succ c \succ a$ .  
 Wähle  $S = \{a, b, c\}$  und OBdA. sei  $f(\succ_1^S, \dots, \succ_n^S) = a$ . Die Transformation zu  $\succ^{S'}$  mit  $S' = \{a, c\}$  zeigt  $f(\succ_1^{S'}, \dots, \succ_n^{S'}) = a$ , und daher  $a \succ c$ . Dies ergibt einen Widerspruch mit Antisymmetrie.

Damit ist gezeigt: Wenn  $f$  anreizkompatibel und ausschöpfend in  $A$  ist, dann ist  $F$  eine gültige soziale Nutzenfunktion. □ [Lemma]

*Beweis (Lemma 9.8).*

- Einstimmigkeit:  $F(\succ, \dots, \succ) = \succ$ .  
 Wenn  $a \succ_i b$  für alle  $i$ , dann gilt mit Behauptung und Monotonie  $f(\succ_1^{\{a,b\}}, \dots, \succ_n^{\{a,b\}}) = a$ .
- IIA  
 Wir nehmen an  $a \succ_i b \Leftrightarrow a \succ'_i b$ . Beachte  $f(\succ_1^{\{a,b\}}, \dots, \succ_n^{\{a,b\}}) = f(\succ'_1^{\{a,b\}}, \dots, \succ'_n^{\{a,b\}})$ , da bei der Transformation von  $\succ_i^{\{a,b\}}$  in  $\succ'_i^{\{a,b\}}$  das Ergebnis gleich bleibt aufgrund von Monotonie und der Behauptung.
- Keine Diktatur  
 Offensichtlich □ [Lemma]

Zusammenfassend haben wir den Satz von Gibbard-Satterthwaite gezeigt, indem wir eine gültige soziale Nutzenfunktion  $F$  aus einer monotonen sozialen Auswahlfunktion  $f$  abgeleitet haben. Des Weiteren können wir zeigen, dass  $F$  keine Diktatur ist, wenn  $f$  keine ist. Mit dem Satz von Arrow folgt ein Widerspruch. □ [Satz]

## 9.2 Strukturierte Präferenzen

Der Satz von Gibbard-Satterthwaite ist entmutigend, aber er setzt voraus, dass die Wähler allgemeine Präferenzen haben können. Wenn die Präferenzen mehr Struktur aufweisen, dann existiert eine umfangreichere Klasse von anreizkompatiblen Auswahlfunktionen.

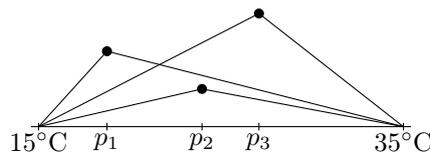
### 9.2.1 Präferenzen mit Scheitelpunkt

Wir betrachten eine Menge von Alternativen, die entlang einer Linie geordnet werden können. Wir nehmen an, dass  $A \subseteq [0, 1]$ .

**Definition 9.10.** Eine Präferenzordnung  $\succ_i$  über  $A$  hat *einen Scheitelpunkt*, wenn es einen Punkt  $p_i \in A$  gibt, so dass für alle  $x \in A \setminus \{p_i\}$  und  $\lambda \in [0, 1)$  gilt

$$(\lambda x + (1 - \lambda)p_i) \succ_i x .$$

Ein Beispiel für das Problem ist, die Raumtemperatur in einem gemeinsamen Büro einzustellen. Jeder Angestellte hat eine optimale Temperatur und möchte, dass die Temperatur so nah wie möglich daran eingestellt wird.



Für Präferenzen mit einem Scheitelpunkt kann man den Satz von Gibbard-Satterthwaite nicht anwenden.

**Algorithmus 18:** Ordnungsmechanismus für Präferenzen mit Scheitelpunkt

---

Sei  $k$  eine Zahl in  $\{1, \dots, n\}$   
 Frage nur nach Scheitelpunkten  $p_1, \dots, p_n$  der Wähler.  
 Sortiere die Scheitelpunkte aufsteigend von 0 nach 1  
 Gib den  $k$ -ten Scheitelpunkt in der Sortierung aus

---

**Proposition 9.11.** Für jedes feste  $k \in \{1, \dots, n\}$  ist der Ordnungsmechanismus anreizkompatibel. Wenn  $n \geq 2$ , dann ist er keine Diktatur.

*Beweis.* Sei  $p$  das Ergebnis, wenn alle Wähler ihren wahren Scheitelpunkt nennen. Wenn  $p_i > p$ , dann kann Wähler  $i$  das Ergebnis nicht durch  $p'_i > p_i$  ändern. Wenn er einen Scheitelpunkt  $p'_i \leq p$  lügt, dann kann nur ein schlechteres Ergebnis  $p' \leq p$  entstehen. Das gleiche Argument kann man für  $p_i < p$  anwenden. Offensichtlich ist der Mechanismus keine Diktatur.  $\square$

Der bekannteste Fall ist der *Median-Mechanismus* mit  $k = \lfloor (n+1)/2 \rfloor$ . Mit dem Durchschnitt der Scheitelpunkte  $\sum_{i=1}^n p_i/n$  wäre der Mechanismus dagegen nicht anreizkompatibel. Mit dem gleichen Argument wie oben bleibt der Ordnungsmechanismus anreizkompatibel wenn – zusätzlich zu den genannten Scheitelpunkten – eine beliebige Anzahl von vorher bekannten und festgelegten Ergebnissen  $y_j \in A$  hinzugenommen und mit sortiert werden. Der Mechanismus wählt dann den  $k$ -größten Scheitelpunkt von  $\{p_1, \dots, p_n, y_1, \dots, y_m\}$ .

Für jedes feste  $k$  ist der Ordnungsmechanismus *anonym*, d.h. er erfüllt

$$f(\succ_1, \dots, \succ_n) = f(\succ'_1, \dots, \succ'_n),$$

wenn  $(\succ_1, \dots, \succ_n)$  eine Permutation von  $(\succ'_1, \dots, \succ'_n)$  ist.

**Satz 9.12** (Moulin 1980; Ching 1997). Seien  $p_i$  die genannten Scheitelpunkte. Eine soziale Auswahlfunktion  $f$  ist anreizkompatibel, surjektiv und anonym für Präferenzen mit einem Scheitelpunkt genau dann, wenn  $f$  ein Ordnungsmechanismus über einer Menge  $\{p_1, \dots, p_n, y_1, \dots, y_m\}$  ist, wobei  $y_j \in [0, 1]$  feste Ergebnisse sind.

Das Resultat ist eine vollständige Charakterisierung für anonyme anreizkompatible soziale Auswahlfunktionen. Anonymität ist für die Charakterisierung notwendig: Eine Diktatur ist kein Ordnungsmechanismus, surjektiv und anreizkompatibel (aber nicht anonym).

### 9.2.2 Hausallokation

Eine andere Struktur ist das Matching mit Präferenzen über Objekte. Diese Variante nennt sich einseitiges Matching. Es werden Spieler zu Häusern zugeteilt, über welche sie Präferenzen besitzen.

- $n$  Spieler und  $n$  Häuser
- Annahme: *Spieler  $i$  besitzt Haus  $i$*   
(nicht unbedingt notwendig, vereinfacht Analyse)
- Spieler  $i$  hat eine Präferenzordnung  $\succ_i$  über Häuser
- Weise jedem Spieler eines der Häuser zu

Die Menge  $A$  der Ergebnisse sind alle *vollständigen Paarungen* von Häusern und Spielern. Ein Spieler hat aber nur eine Präferenz für *das Haus, das er erhält*. Damit sind alle Paarungen, in denen Spieler  $i$  das gleiche Haus bekommt, für ihn gleichwertig.

Was sind anreizkompatible Mechanismen mit “guten” Eigenschaften?

Sei  $G = (V, E)$  der gerichtete Graph:

- $V$  Menge verbleibender Spieler mit ihren Häusern
- $E$  Menge von gerichteten Kanten:

$$(i, \ell) \in E \Leftrightarrow \ell \in V \text{ besitzt bestes verbleibendes Haus für } i \in V$$

---

**Algorithmus 19:** Top-Trading-Cycles (TTC) Mechanismus

---

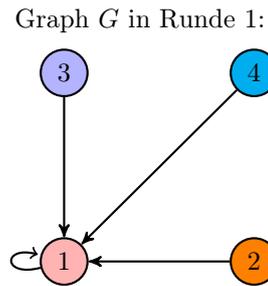
Frage Präferenzen der Spieler ab  
**while**  $V \neq \emptyset$  **do**  
    Erstelle Kantenmenge  $E$  wie oben beschrieben  
    Berechne alle gerichteten Kreise  $C_1, \dots, C_h$  in  $G$   
    (Schleifen gelten als Kreise, Kreise disjunkt)  
    **for** jede Kante  $(i, \ell)$  in jedem Kreis  $C_1, \dots, C_h$  **do**  
        Weise Haus  $\ell$  an Spieler  $i$  zu  
        Entferne alle Spieler in  $C_1, \dots, C_h$  aus  $V$

---

**Beispiel 9.13.** *TTC*

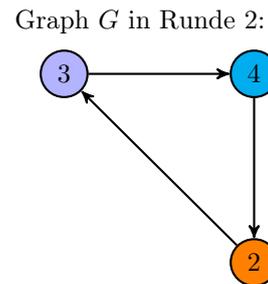
Zu Beginn ist kein Spieler zu einem Haus zugeordnet. Im Graphen wird für jeden Spieler eine Kante zu seinem besten verbleibenden Haus eingefügt.

| Spieler | Präferenzordnung |
|---------|------------------|
| 1       | 1 2 3 4          |
| 2       | 1 3 2 4          |
| 3       | 1 4 2 3          |
| 4       | 1 2 4 3          |



Zuweisung: Spieler 1 – Haus 1

| Spieler | Präferenzordnung |
|---------|------------------|
| 2       | 3 2 4            |
| 3       | 4 2 3            |
| 4       | 2 4 3            |



Zuweisung: Spieler 2 – Haus 3, Spieler 3 – Haus 4, Spieler 4 – Haus 2



**Beobachtung:**

- Jeder Spieler in  $G$  hat Ausgangsgrad 1. Es gibt immer mindestens einen Kreis in  $G$ , und alle Kreise in  $G$  sind disjunkt.
- Seien  $V_k$  die Spieler, die in der  $k$ -ten Runde von TTC entfernt werden. Jeder Spieler in  $V_k$  bekommt das Haus, das er am liebsten mag – abgesehen von Häusern der Spieler in  $V_1 \cup \dots \cup V_{k-1}$ .
- Spieler  $i \in V_k$  bekommt sein bestes Haus in der  $k$ -ten Runde. Der Besitzer dieses Hauses liegt auch in  $V_k$  und erhält somit auch sein bestes Haus.

**Satz 9.14** (Roth 1982). *Der TTC Mechanismus ist anreizkompatibel.*

*Beweis.* Betrachte Spieler  $i \in V_j$  mit wahrer Präferenz  $\succ_i$ . Wenn er die Wahrheit sagt, bekommt er das beste verbleibende Haus in Runde  $j$ . Betrachte ein Haus von einem Spieler in  $V_1 \cup \dots \cup V_{j-1}$ , das ihm besser gefällt. Spieler  $i$  kann keines dieser Häuser bekommen:

- In Runde  $k = 1, \dots, j - 1$  will kein Spieler  $\ell \in V_k$  das Haus von  $i$  haben, sonst wäre  $i$  mit  $\ell$  in einem Kreis.
- Kein Spieler  $\ell \in V_k$  möchte das Haus von  $i$  in einer Runde  $< k$  haben, sonst würde  $\ell$  es in Runde  $k$  immer noch haben wollen.

Die Häuser der Spieler in  $V_1 \cup \dots \cup V_{j-1}$  bleiben unerreichbar, egal welche Präferenzordnung Spieler  $i$  nennt. Wenn  $i$  aber bis Runde  $j$  warten muss, dann liefert TTC bei ehrlicher Bewertung das beste (verbleibende) Haus.  $\square$

TTC ist also ein anreizkompatibler Mechanismus, aber davon gibt es mehrere. Warum ist TTC besser als andere Ansätze?

Sei  $M$  eine Zuweisung von Häusern – Spieler  $i$  bekommt Haus  $M(i)$ . Sei  $M_S$  eine Zuweisung, die aus  $M$  entsteht, wenn die Koalition  $S \subseteq \mathcal{N}$  ihre initialen Häuser anders unter sich aufteilen.

**Definition 9.15.** Eine Spielermenge  $S \subseteq \mathcal{N}$  bildet eine *blockierende Koalition* für  $M$ , wenn es eine Zuweisung  $M_S$  gibt, so dass

- für jeden Spieler  $j \in C$  ist  $M_C$  genauso gut:  $M_C(j) \succeq_j M(j)$
- für mind. einen Spieler  $i \in C$  ist  $M_C$  besser:  $M_C(i) \succ_i M(i)$

Eine Zuweisung  $M$  ohne blockierende Koalition liegt im *Kern* des Spiels.

Eine Kernzuweisung hat eine Optimalitätseigenschaft: Keine Teilmenge von Spielern möchte ihre Häuser vom Mechanismus zurückhalten und anders (unter sich) verteilen. Insbesondere bekommt jeder Spieler  $i$  ein Haus, das er mindestens so gerne mag wie sein anfängliches Haus  $i$ .

**Satz 9.16** (Roth, Postlewaite 1977). *Der TTC Mechanismus berechnet die eindeutige Zuweisung im Kern des Spiels.*

*Beweis.* Induktion: Nur die TTC-Zuweisung kann im Kern sein, aber keine andere.

**Induktionsanfang:** Jeder  $i \in V_1$  bekommt sein allerbestes Haus.  $V_1$  ist also eine blockierende Koalition für jede Zuweisung, die die Häuser in  $V_1$  nicht so verteilt wie TTC.

**Induktionsannahme:** Die Häuser der Spieler in  $V_1, \dots, V_{j-1}$  müssen verteilt werden wie bei TTC.

**Induktionsschritt:** Unter der Annahme bekommt jeder  $i \in V_j$  das beste (verbleibende) Haus.  $V_j$  ist also eine blockierende Koalition für jede Zuweisung, die die Häuser in  $V_1, \dots, V_{j-1}$  so verteilt wie TTC, aber  $V_j$  nicht so verteilt wie TTC.

Damit gilt: Entweder die TTC-Zuweisung ist im Kern oder der Kern ist leer. Sei nun  $M$  die TTC-Zuweisung und  $M_S$  eine Umverteilung der Häuser der Spieler  $S \subseteq M$ . Spieler  $i \in S$  hat vorher Haus  $i$ . Wenn  $S$  mit  $M_S$  blockierend sein soll, dann muss  $i$  nachher ein Haus erhalten, sonst wäre er schlechter dran als in  $M$ . Damit bildet die Umverteilung in  $S$  eine Menge von Kreisen:

- Kreis enthält Spieler von  $N_j$  und  $N_\ell$  mit  $j < \ell$ :  
Mindestens ein Spieler  $i \in N_j$  bekommt ein Haus von einem Spieler in  $N_\ell$  und ist damit schlechter dran als in  $M$ .

- Kreis enthält nur Spieler einer Menge  $N_j$ :  
In  $M$  bekommt jeder Spieler  $i \in N_j$  sein bestes Haus unter den Häusern der Spieler in  $N_j$ .  
Mindestens ein Spieler ist schlechter dran als in  $M$ .

Also ist die TTC-Zuweisung im Kern des Spiels, und somit die eindeutige Zuweisung mit dieser Eigenschaft.  $\square$

Ein anderer Mechanismus ist der Random Serial Dictatorship Mechanismus. Die Anfangsannahme, dass Spieler  $i$  mit Haus  $i$  startet, wird nicht benötigt. Dieser Mechanismus ist eine geordnete Variante der Diktatur. Das folgende Resultat kann man

---

**Algorithmus 20:** Random Serial Dictatorship (RSD) Mechanismus

---

```

Ordne Spieler in zufälliger Reihenfolge
Frage Präferenzen der Spieler ab
Sei  $V$  die Menge aller Häuser
for  $i = 1, 2, 3, \dots, n$  do
  | Weise Spieler  $i$  sein bestes Haus  $h$  aus  $V$  zu
  | Entferne  $h$  aus  $V$ 

```

---

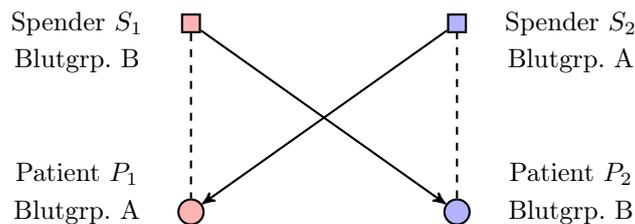
ähnlich wie für TTC beweisen.

**Satz 9.17.** Für jede gewählte Ordnung ist der RSD Mechanismus anreizkompatibel.

Liegt die Zuweisung von RSD im Kern? Für jede Ordnung? Für keine? Für einige?

### 9.3 Nierenaustausch

In vielen Ländern gibt es mittlerweile (Pläne für) Programme zum Nierentausch. Patienten haben oft einen Bekannten/Verwandten als Lebendspender, dessen Niere (z.B. aufgrund von Blutgruppe) nicht zum Patienten passt. Das Ziel ist ein *Organaustausch*: Zwei inkompatible (Patient,Spender)-Paare tauschen die Spenderorgane aus, wenn sie zum jeweils anderen Patienten passen.



Im Prinzip also eine Hausallokation:

- “Häuser” sind Spenderorgane, “Spieler” sind Patienten
- Präferenzen über Organe = W.keit der erfolgreichen Transplantation.
- TTC Mechanismus ist anreizkompatibel und im Kern des Spiels

Anmerkungen:

Es gibt keine legale Verpflichtung zur Spende. Alle Operationen eines Kreises werden *simultan* durchgeführt, da sonst ein *Anreizproblem* entsteht: Spender  $S_i$  tritt zurück, sobald Patient  $P_i$  seine Niere erhalten hat.

Lange Kreise sind schwierig, da *lange Kreise = viele simultane Operationen*. Wenn dagegen eine *reine Nierenspende* dazukommt, wird der Kreis zum Pfad – dann wird immer erst die Niere von

Spender  $S_i$  entnommen bevor Patient  $P_i$  seine Spende bekommt.  
 Sehr oft gibt es eher *binäre Präferenzen*: Die Niere passt zum Patienten oder nicht.

Wir verfolgen einen Ansatz mit Matching in einem Graphen  $G = (V, E)$ :  
 Für Patient  $P_i$  sei  $E_i$  die Menge von kompatiblen Spendern. Jedes Paar  $(P_i, S_i)$  ist ein *Knoten*  $v_i \in V$ . Wir suchen *einfache Austausche* bzw. Kreise der Länge 2, wobei *Kante*  $\{v_i, v_j\} \in E$ , wenn  $S_i \in E_j$  und  $S_j \in E_i$ . Die Patienten können bei der Nennung der Menge  $E_i$  lügen. Offensichtlich lohnt sich nur ein  $F_i \subseteq E_i$  zu lügen.

---

**Algorithmus 21:** Matching-Mechanismus für Nierentausch
 

---

Frage die Mengen  $F_i$  der Patienten ab  
 Erstelle Graph  $G$  wie oben, wobei  $E = \{\{v_i, v_j\} \mid S_i \in F_j, S_j \in F_i\}$   
 Berechne ein Matching  $M$  von  $G$  mit größter Kardinalität  
 Nierentausch gemäß der Kanten in  $M$

---

Das maximum Matching ist *nicht eindeutig*. Verschiedene maximum Matchings verteilen die Nieren an *unterschiedliche Patienten*. Wir müssen das Matching auf eine "monotone" Art wählen. Dafür *priorisieren* wir die Patienten. Das ist ein üblicher Ansatz, z.B. bei der Erstellung von Wartelisten für die Organspende.

---

**Algorithmus 22:** Maximum Matching mit Prioritäten
 

---

Sei  $M_0$  die Menge aller maximum Matchings von  $G$   
**for**  $i = 1, 2, 3, \dots, n$  **do**  
 Sei  $Z_i$  die Menge von Matchings in  $M_{i-1}$ , in denen Knoten  $v_i$  gematcht ist  
**if**  $Z_i \neq \emptyset$  **then**  
    $\perp$   $M_i \leftarrow Z_i$   
**else**  
    $\perp$   $M_i \leftarrow M_{i-1}$   
**return** beliebiges Matching aus  $M_n$

---

**Satz 9.18.** *Der Matching-Mechanismus mit Prioritäten für Nierenaustausch ist anreizkompatibel.*

Durch eine geschickte implizite Verwaltung der maximum Matchings gibt es eine Implementation für den Matching-Mechanismus mit Prioritäten mit polynomieller Laufzeit.

## 9.4 Stabiles Matching

Stabiles Matching kann man als Spiel aber auch aus der Perspektive der Mechanismen betrachten.

Die Frauen und Männer suchen jeweils einen Partner. Jeder Spieler besitzt eine Präferenzordnung über die Spieler der anderen Seite mit der Ordnung: (besten Partner, zweitbesten, ...). Gesucht ist ein Matching, das den Präferenzen möglichst entspricht.

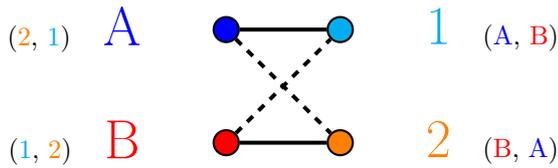
**Definition 2.31.** [Matching mit Präferenzen] Es existiert eine Menge  $\mathcal{X}$  von  $n$  Männern und eine Menge  $\mathcal{Y}$  von  $n$  Frauen. Jeder Mann  $x \in \mathcal{X}$  hat eine *Präferenzordnung*  $\succ_x$  über alle  $y \in \mathcal{Y}$  (analog für die Frauen), dabei ist jeder lieber verpartnert als allein. Für ein Matching  $M$  sei  $M(x) \in \mathcal{Y}$  die Partnerin von Mann  $x \in \mathcal{X}$  in  $M$ , und  $M(y) \in \mathcal{X}$  der Partner von Frau  $y \in \mathcal{Y}$  in  $M$ . Sei  $M(x) = *$ , wenn  $x$  allein in  $M$ , und  $M(y) = *$  genauso.

Wann ist ein Matching stabil? Was ist eine Gefahr für Stabilität?

**Definition 2.32.** [Stabiles Matching] In  $M$  ist  $\{x, y\}$  ein *blockierendes Paar* genau dann, wenn  $x$  und  $y$  sich gegenseitig lieber mögen als ihre jeweiligen Partner  $y' = M(x)$  und  $x' = M(y)$ . Ein Matching  $M$  ohne blockierendes Paar ist ein *stabiles Matching*.

**Satz 2.38.** [Gale, Shapley 1962] *Der Deferred-Acceptance Algorithmus berechnet ein stabiles Matching nach höchstens  $n^2$  Iterationen.*

Der Deferred-Acceptance Algorithmus ist unterspezifiziert: Es wird nicht gesagt, welcher Mann als nächstes einen Antrag macht. Gibt es *mehrere stabile Matchings*? Kann der Algorithmus *mehrere (alle?) stabile Matchings berechnen*? Stabile Matchings sind nicht eindeutig. In diesem Fall gibt es ein *männeroptimales* Matching (durchgezogen) und ein *frauenoptimales* (gestrichelt), bei dem jede(r) gleichzeitig den(die) beste(n) Partner(in) bekommt.



Sei  $h(x) \in \mathcal{Y}$  die beste Frau, so dass ein stabiles Matching  $M'$  mit  $M'(x) = h(x)$  existiert. Sei  $h(y) \in \mathcal{X}$  der beste Mann, so dass ein stabiles Matching  $M'$  mit  $M'(y) = h(y)$  existiert.

**Definition 9.19.** Ein stabiles Matching  $M$  ist

- *männeroptimal*, wenn  $M(x) = h(x)$  für alle  $x \in \mathcal{X}$ .
- *frauenoptimal*, wenn  $M(y) = h(y)$  für alle  $y \in \mathcal{Y}$ .

In einem optimalen Matching bekommen *alle Spieler einer Seite gleichzeitig* den besten Partner, die sie *in irgendeinem stabilen Matching* bekommen können. Es ist gar nicht klar, ob so etwas überhaupt möglich ist.

Der folgende Satz beweist dies, und sogar noch mehr: Egal in welcher Reihenfolge die Anträge im Algorithmus gemacht werden, es kommt immer ein eindeutiges optimales Matching heraus.

**Satz 9.20.** *Der Deferred-Acceptance Algorithmus mit Männerantrag berechnet immer genau das männeroptimale stabile Matching und mit Frauenantrag das frauenoptimale.*

*Beweis.* Mit Widerspruch zur Stabilität:

Sei  $M$  das Matching des Algorithmus mit Männerantrag. Sei  $M'$  stabiles Matching, in dem ein Mann  $j \in \mathcal{X}$  eine strikt bessere Partnerin  $M'(j) \succ_j M(j)$  bekommt.

Da  $M'(j) \succ_j M(j)$ , gibt es eine Iteration im Algorithmus, in der  $j$  der Frau  $M'(j)$  einen Antrag gemacht hat und abgelehnt wird. Eventuell gibt es noch mehr Iterationen im Algorithmus, in denen ein Mann  $x$  von der Partnerin  $M'(x)$  in  $M'$  abgelehnt wird. Betrachte die erste dieser Iterationen. Frau  $M'(x)$  lehnt Mann  $x$  nur ab, weil sie einen besseren Antrag von Mann  $i \neq x$  bekommt (d.h.  $i \succ_{M'(x)} x$ ). Da dies die allererste Ablehnung dieser Art ist, mag Mann  $i$  die Frau  $M'(x)$  mehr als  $M'(i)$  (d.h.  $M'(x) \succ_i M'(i)$ ). Also ist  $(i, M'(x))$  ein blockierendes Paar in  $M'$  und  $M'$  kein stabiles Matching – Widerspruch.  $\square$

**Satz 9.21.** *Der Deferred-Acceptance Algorithmus mit Männerantrag ist anreizkompatibel für die Männer.*

*Beweis.* Wir vereinbaren folgende Notation. Ein Lügner habe oBdA die Nummer 1.

- Echtes Profil:  $\pi = (\succ_1, \dots, \succ_n)$ , Algorithmus berechnet  $M$
- Mann 1 lügt:  $\pi' = (\succ', \succ_2, \dots, \succ_n)$ , Algorithmus berechnet  $M'$

Wenn Lügen profitabel ist, dann gilt  $M'(1) \succ_1 M(1)$ . Wir werden zeigen, dass in diesem Fall  $M'$  nicht stabil ist für  $\pi'$  – ein Widerspruch.

Die folgende Behauptung zeigt, dass in der Menge der Männer, die von der Lüge profitieren, die zugewiesenen Partnerinnen *untereinander ausgetauscht werden*.

**Behauptung 9.22.** Seien  $R = \{x \mid M'(x) \succ_x M(x)\}$  die Männer, die in  $M'$  profitieren. Für jeden Mann  $x \in R$  und seine neue Partnerin  $y = M'(x)$  ist der Partner  $x' = M(y)$  in  $M$  auch  $x' \in R$ .

*Beweis (Behauptung).*

- $x' = 1$ : Klar,  $x' \in R$  per Annahme, dass 1 lügen möchte.
- $x' \neq 1$ : Da  $x \in R$  gilt  $y \succ_x M(x)$ . Dann  $x' \succ_y x$ , denn sonst hätte  $M$  das blockierende Paar  $\{x, y\}$ . Also gilt  $M'(x') \succ_{x'} y$ , denn sonst hätte  $M'$  das blockierende Paar  $\{x', y\}$ . Es folgt  $M'(x') \succ_{x'} M(x')$  und somit  $x' \in R$ . □ (Behauptung)

Die Menge der Partnerinnen von profitierenden Männern ist also in beiden Matchings  $M$  und  $M'$  die gleiche:  $T = \{y \mid M(y) \in R\} = \{y \mid M'(x) \in R\}$ .

Da alle Männer aus  $R$  besser gematcht sind in  $M'$ , sind alle Frauen aus  $T$  schlechter gematcht, sonst gäb es ein blockierendes Paar in  $M$ .

Betrachte nun die Berechnung von  $M$  mit dem echten Profil  $\pi$ :

Sei  $x$  der letzte Mann von  $R$ , der einen Antrag macht. Dieser Antrag geht an seine endgültige Partnerin  $y = M(x) \in T$ . Es gibt keine weiteren Anträge von Männern in  $R$ , daher sind alle außer  $x$  schon gematcht wie in  $M$ . Also:  $y$  hat  $M'(y)$  in einer Runde davor abgelehnt.  $y$  kann nur einen Antrag eines Mannes  $x' \notin R$  haben mit  $x' \succ_y M'(y)$ .  $x' \notin R$  bedeutet  $M(x') \succeq_{x'} M'(x')$ , und Ablehnung von  $y$  bedeutet  $y \succ_{x'} M(x')$ . Somit  $y \succ_{x'} M'(x')$  und  $x' \succ_y M'(y)$ . Da  $x' \notin R$ , gilt  $x' \neq 1$ . Also hat  $M'$  ein blockierendes Paar für  $\pi'$ , ein Widerspruch. □ (Satz)

Der Deferred-Acceptance Algorithmus mit Frauenantrag ist nicht anreizkompatibel für Männer:

**Beispiel 9.23.**

|   |              |              |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
|---|--------------|--------------|---|---|---|---|---|---|---|---------|---|---|--------------|--|--------------|--------------|--|---|---|---|---|---|---|--------------|--------------|---|--------------|---|---|
| <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">B</td> <td style="padding: 2px;"><del>A</del></td> <td style="border: 1px solid black; padding: 2px;">A</td> </tr> <tr> <td style="padding: 2px;">A</td> <td style="border: 1px solid black; padding: 2px;">C</td> <td style="padding: 2px;">B</td> </tr> <tr> <td style="padding: 2px;">C</td> <td style="padding: 2px;">B</td> <td style="padding: 2px;">C</td> </tr> <tr style="border-top: 1px solid black;"> <td style="padding: 2px;">Frauen:</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">2</td> <td style="padding: 2px;">3</td> </tr> </table>   | B            | <del>A</del> | A | A | C | B | C | B | C | Frauen: | 1 | 2 | 3            | <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><del>B</del></td> <td style="padding: 2px;"><del>A</del></td> <td style="padding: 2px;"><del>A</del></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">A</td> <td style="border: 1px solid black; padding: 2px;">C</td> <td style="border: 1px solid black; padding: 2px;">B</td> </tr> <tr> <td style="padding: 2px;">C</td> <td style="padding: 2px;">B</td> <td style="padding: 2px;">C</td> </tr> <tr style="border-top: 1px solid black;"> <td style="padding: 2px;">1</td> <td style="padding: 2px;">2</td> <td style="padding: 2px;">3</td> </tr> </table> | <del>B</del> | <del>A</del> | <del>A</del>   | A | C | B | C | B | C | 1            | 2            | 3 |              |   |   |
| B   | <del>A</del> | A            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| A   | C            | B            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| C   | B            | C            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| Frauen:   | 1            | 2            | 3 |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| <del>B</del>  | <del>A</del> | <del>A</del> |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| A   | C            | B            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| C   | B            | C            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| 1   | 2            | 3            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| <table style="margin: auto; border-collapse: collapse;"> <tr style="border-top: 1px solid black;"> <td style="padding: 2px;">Männer:</td> <td style="padding: 2px;">A</td> <td style="padding: 2px;">B</td> <td style="padding: 2px;">C</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">3</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"><del>2</del></td> <td style="padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="padding: 2px;"></td> </tr> </table> | Männer:      | A            | B | C | 1 | 3 | 1 |   | 3 | 1       | 3 |   | <del>2</del> | 2  | 2            |              | <table style="margin: auto; border-collapse: collapse;"> <tr style="border-top: 1px solid black;"> <td style="padding: 2px;">A</td> <td style="padding: 2px;">B</td> <td style="padding: 2px;">C</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">3</td> <td style="padding: 2px;">1</td> </tr> <tr> <td style="padding: 2px;"><del>2</del></td> <td style="padding: 2px;"><del>1</del></td> <td style="padding: 2px;">3</td> </tr> <tr> <td style="padding: 2px;"><del>3</del></td> <td style="padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">2</td> </tr> </table> | A | B | C | 1 | 3 | 1 | <del>2</del> | <del>1</del> | 3 | <del>3</del> | 2 | 2 |
| Männer:   | A            | B            | C |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| 1   | 3            | 1            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| 3   | 1            | 3            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| <del>2</del>  | 2            | 2            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| A   | B            | C            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| 1   | 3            | 1            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| <del>2</del>  | <del>1</del> | 3            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |
| <del>3</del>  | 2            | 2            |   |   |   |   |   |   |   |         |   |   |              |  |              |              |  |   |   |   |   |   |   |              |              |   |              |   |   |

■

## 9.5 Zusammenfassung

Für Wahlen haben wir zwei ernüchternde Ergebnisse gezeigt. Aus dem Satz von Arrow und dem Satz von Gibbard-Satterthwaite folgt, dass im Allgemeinen ohne Einschränkung der Präferenzen jede anreizkompatible Wahl einer Diktatur entspricht. In vielen Fällen weisen die Präferenzen der Wähler jedoch eine Struktur auf, weshalb die gezeigten Sätze nicht mehr anwendbar sind. Für Präferenzen mit Scheitelpunkt existieren die sehr einfachen und anreizkompatiblen Ordnungsmechanismen und für Hausallokation liefert TTC eine eindeutige Zuweisung der Häuser zu Spielern, so dass sich kein Spieler durch seine Teilnahme verschlechtert. Ebenso wurden anreizkompatible

Mechanismen für Nierenaustausch und stabiles Matching besprochen.

Zur Vertiefung wird folgende Literatur empfohlen:

- Roth, Postlewaite. Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics* 4(2):131–137, 1977.
- Roth, Sönmez, Ünver. Pairwise kidney exchange. *Journal of Economic Theory* 125(2):151–188, 2005.
- Gale, Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly* 69:9–14, 1962.
- Dubins, Freedman. Machiavelli and the Gale-Shapley Algorithm. *American Mathematical Monthly* 88(7):485–494, 1981.
- Roth. *The Economics of Matching: Stability and Incentives*. *Mathematics of Operations Research* 7(4):617–628, 1982.
- Brandt, Conitzer, Endriss, Lang, Procaccia. *Handbook of Computational Social Choice* (Kapitel 14)
- Nisan, Vazirani, Roughgarden, Tardos. *Algorithmic Game Theory*, 2007. (Kapitel 9 und 10)
- Roughgarden. *Twenty Lectures on Algorithmic Game Theory*, 2016. (Kapitel 9.4 und 10)