Eine spieltheoretische Betrachtung des Pokerspiels

Seminar TUD Computer Poker Challenge

Stefan Lück & Claudio Weck

Agenda

Spieltheoretische Grundlagen

Das Pokerspiel

Verbesserte Lösungsstrategien

Die Poker Challenge

Spieltheoretische Grundlagen

Was ist Spieltheorie

- Die Spieltheorie analysiert das Verhalten bei Entscheidungssituationen mit *mehreren interagierenden Akteuren*

- Erstmals formal beschrieben durch John von Neumann

- Ziel: Methoden zur Modellierung und Lösung von Spielen

- Definition eines Spiels:

$$
\begin{array}{lll}
\mathcal{P} & & \textit{Menge an Spielern} \\
\mathcal{A}_p & .\forall p \in \mathcal{P} & \textit{Handlungsalternativen pro Spieler} \\
u_p : \mathcal{A}_1 \times \cdots \times \mathcal{A}_{|\mathcal{P}|} \to \mathbb{R} & .\forall p \in \mathcal{P} & \textit{Auszahlungsfunktion pro Spieler}
\end{array}
$$

Beispiel: Schere-Stein-Papier

- Zwei Spieler
- Jeweils 3 Handlungsalternativen
- Auszahlungen +1, -1 oder 0

- Darstellung in Normalform:

		Spieler 2		
		Schere	Stein	Papier
Spieler 1	Schere	0 ; 0	-1 ; 1	1 ; -1
	Stein	1 ; -1	0 ; 0	-1 ; 1
	Papier	-1 ; 1	1 ; -1	0 ; 0

Klassifikation von Spielen

- Statische vs. Dynamische (rundenbasierte) Spiele
- Zufällige vs. Deterministische Spiele
- Vollkommene vs. Unvollkommene Spiele

	ohne Zufall	mit Zufall
vollkommen	Schach Go	Backgammon Kniffel
nicht vollkommen	Schiffe versenken	Poker Rommé

- Weitere Eigenschaften:
  - Kompetitave Spiele (→ Nullsummenspiele)
  - Symmetrische Spiele

Spielvorhersage - Strategien

- Def. Spielstrategie: Gibt Handlung für jede denkbare Situation vor
- Def. Gesamtstrategie: Zusammenfassung aller Einzelstrategien

$$S = \{s_p | \forall p \in \mathcal{P}\}$$

  - → bestimmt eindeutig den Spielverlauf
- Def. Fremdstrategien: Zusammenfassung aller Mitspielerstrategien

$$S_{-p} = \{s_q | \forall q \in \mathcal{P} \wedge q \neq p\}$$

- Def. Gemischte Strategien: Wahrscheinlichkeitsverteilung über Handlung

$$s_p \in \mathcal{S}_p = \left\{ \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_{|\mathcal{A}_p|} \end{pmatrix} \mid \sum_{i=1}^{|\mathcal{A}_p|} \pi_i = 1 \ \wedge \ \forall \pi_i : \pi_i \geq 0, \ i \in \{1, \ldots, |\mathcal{A}_p|\} \right\}$$

Spielvorhersage - Lösungen

- Das Rationalitätsprinzip: Jeder will seinen Nutzen maximieren!

$$s_p^* = \arg\max_{s_p \in \mathcal{S}_p} u_p(s_p | S_{-p}) \quad .\forall p \in \mathcal{P}$$

- Gleichgewichtsstrategien: Die gegenseitig beste Antwort → Lösung!

$$S^* \rightarrow \forall p \in \mathcal{P}. \quad s_p^* = \arg\max_{s_p \in \mathcal{S}_p} u_p(s_p | S_{-p}^*)$$

- Erwartete Auszahlung bei gemischten Strategien

$$u_p(s_p | S_{-p}) = E\left(u_p | S_{-p}\right) = \sum_{i=1}^{|\mathcal{A}_p|} \pi_i \cdot u_p(a_{p,i} | S_{-p})$$

Beispiel - Nashgleichgewicht

- Betrachte das Spiel Schere-Stein-Papier

		Spieler 2		
		Schere	Stein	Papier
Spieler 1	Schere	0 ; 0	-1 ; 1	1 ; -1
	Stein	1 ; -1	0 ; 0	-1 ; 1
	Papier	-1 ; 1	1 ; -1	0 ; 0

- Keine optimale reine Strategie
- Aber optimale gemischte Strategie für alle (symm.):
  $s^* = (\frac{1}{3}$ Schere; $\frac{1}{3}$ Stein; $\frac{1}{3}$ Papier$)$
- Erwartete Auszahlung: $3 \times (\frac{1}{3})^2 \times 1 - 3 \times (\frac{1}{3})^2 \times 1 - 3 \times (\frac{1}{3})^2 \times 0 = 0$
- Leider auch nie besser…

Das Pokerspiel

Klassifikation von Poker - Erinnerung

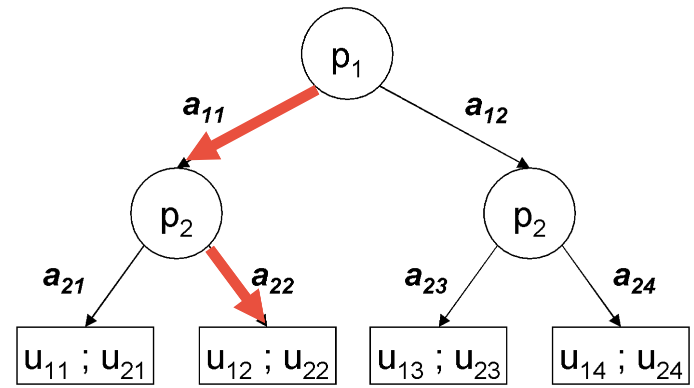
- Poker ist ein
  - Dynamisches
  - Unvollkommenes
  - Zufallsbeeinflusstes
  - Nullsummen-
  - Spiel

- Wie kann man das darstellen?

- Wie kann man es lösen?

Extensive Spielform

- Darstellung dynamischer Spiele intuitiv durch Spielbäume



- Elemente eines Spielbaums:
  - Knoten: symbolisieren Entscheidungs- situationen der Spieler
  - Kanten: stellen Handlungsalternativen dar
  - Blätter: repräsentieren das Spielende mit Auszahlungen

- Spielablauf:
  - Spiel beginnt immer an der Wurzel
  - Entscheidung stellt Verzweigung im Baum dar
  - Spielinstanz: Ein Pfad von der Wurzel in ein Blatt

Extensive Spielform - Erweiterung

TECHNISCHE UNIVERSITÄT DARMSTADT

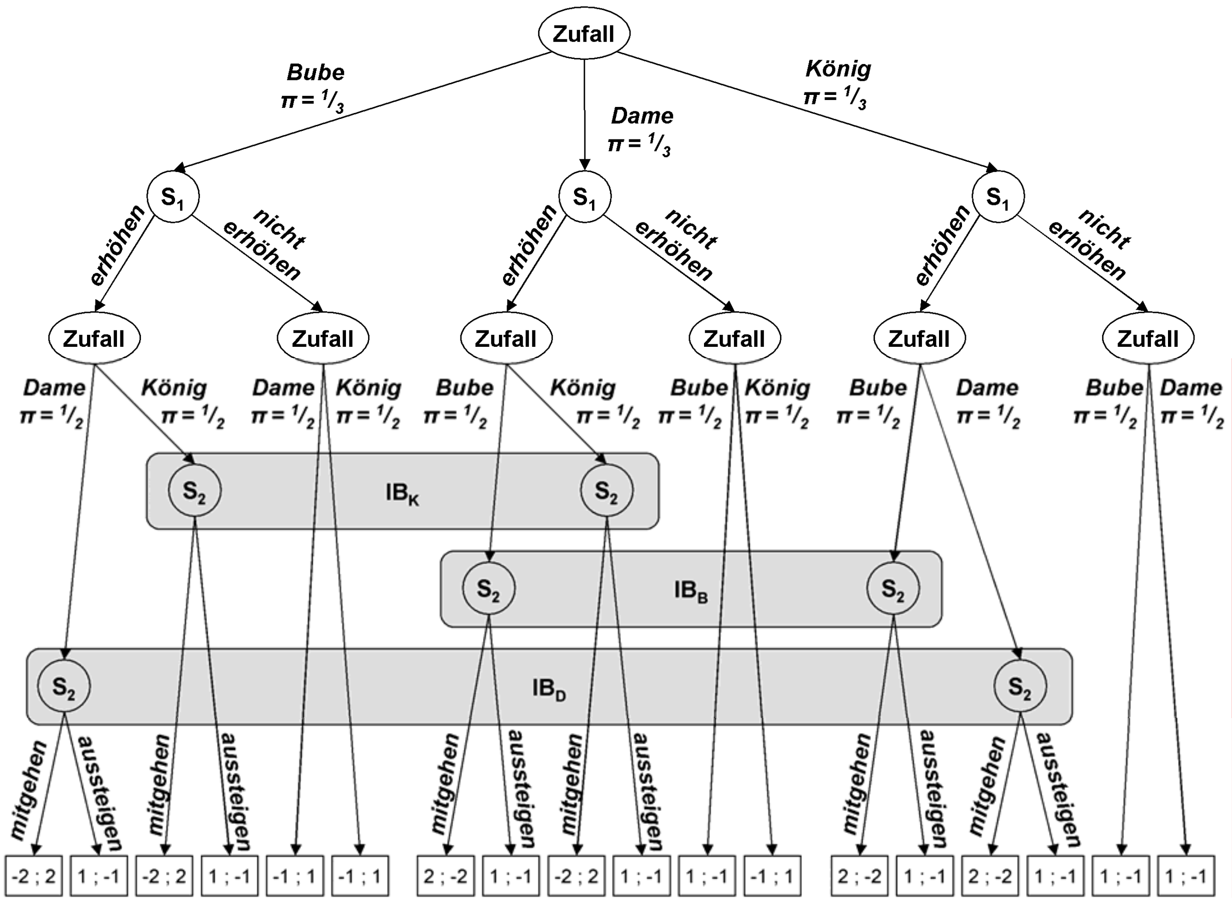
- Modellieren des Zufalls
  - Durch einen zusätzlichen Spieler Natur
  - Feste (gemischte) Strategie in Form einer Wahrscheinlichkeitsverteilung

- Modellierung von unvollkommenen Informationen
  - Spieler fehlt Informationen über den aktuellen Spielstatus
    → Er weiß daher nicht genau, an welchen Knoten er sich befindet!
  - Modelliert durch Informationsbezirke
    - Gruppierung von Knoten genau eines Spielers
      mit identischen Handlungsmöglichkeiten
    - Stellt nur eine einzelne logische Entscheidung für den Spieler dar!

Spielbaum-Beispiel: Einfach-Poker

- Stark vereinfachtes Poker:
  - Zwei Spieler
  - Nur drei Karten (B,D,K)
  - Grundeinsatz 1 GE
  - Spieler 1 darf auf 2 GE erhöhen
  - Spieler 2 entscheidet bei Erhöhung, ob er mitgeht
  - Dann Aufdeckung

Spielbaum-Beispiel: Einfach-Poker



Lösungsverfahren

- Exakte Lösung (Gleichgewicht) existiert [Nash]

- Exakte Lösungsverfahren existieren ebenfalls

- Wir stellen zwei vor:
  - Rückwärtsinduktion mit Eliminierung dominanter Strategien
  - MaxiMin-Methode

Rückwärtsinduktion

- Lösung für Spiele mit vollkommener Information
- Basiert auf dem Antizipieren der Entscheidungen der Mitspieler
- Verfahren:
  - Interpretiere Subbäume als Teilspiele
  - Löse alle Teilspiele der untersten Ebene und ersetze sie durch eine Auszahlung nach dem Rationalitätsprinzip
  - Wiederhole den Vorgang, bis der Baum auf ein einzelnes Blatt reduziert ist
- Leider nicht direkt anwendbar für Poker (Informationsbezirke!)

Eliminieren dominierter Strategien

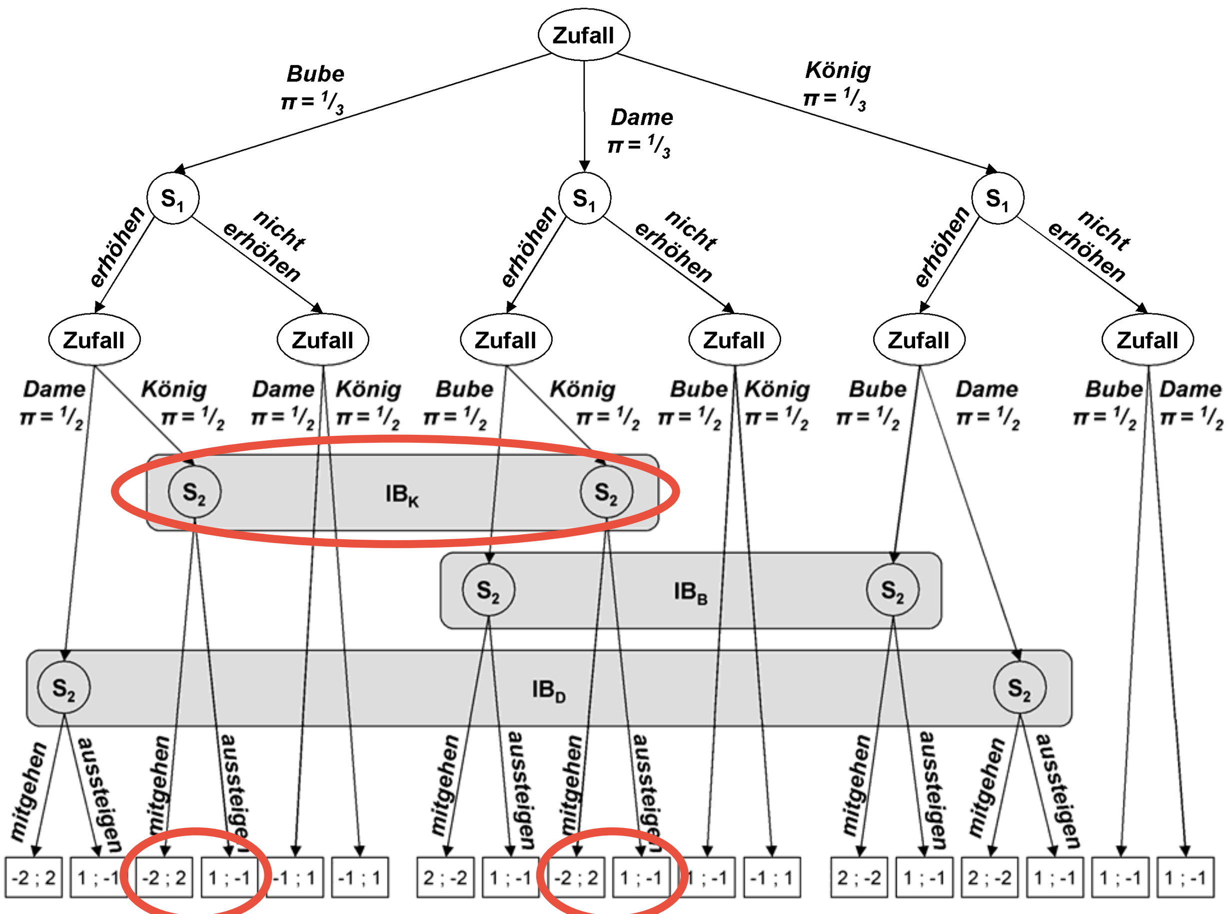
- Idee: Untersuche Informationsbezirke auf Strategien, die immer besser sind als andere.

- Formal: Stratetgie a_i wird durch a_j dominiert, falls:

$$\exists a_j \in \mathcal{A}_p . u_p(a_i | w_b) < u_p(a_j | w_b) \forall w_b \in B$$

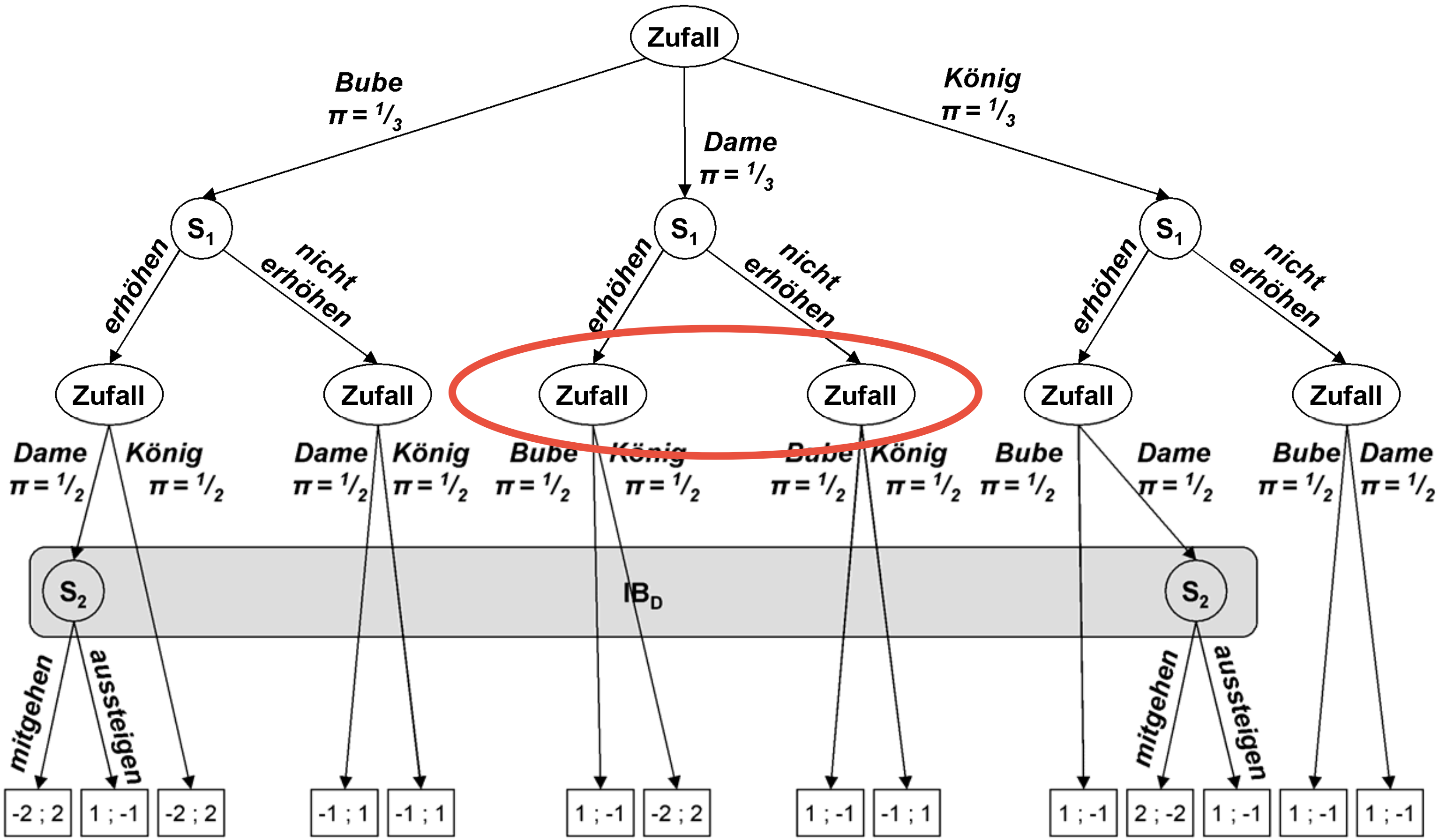
- Beispiel am einfachen Poker

Rückwärtsinduktion und dominierte Strategien → Beispiel

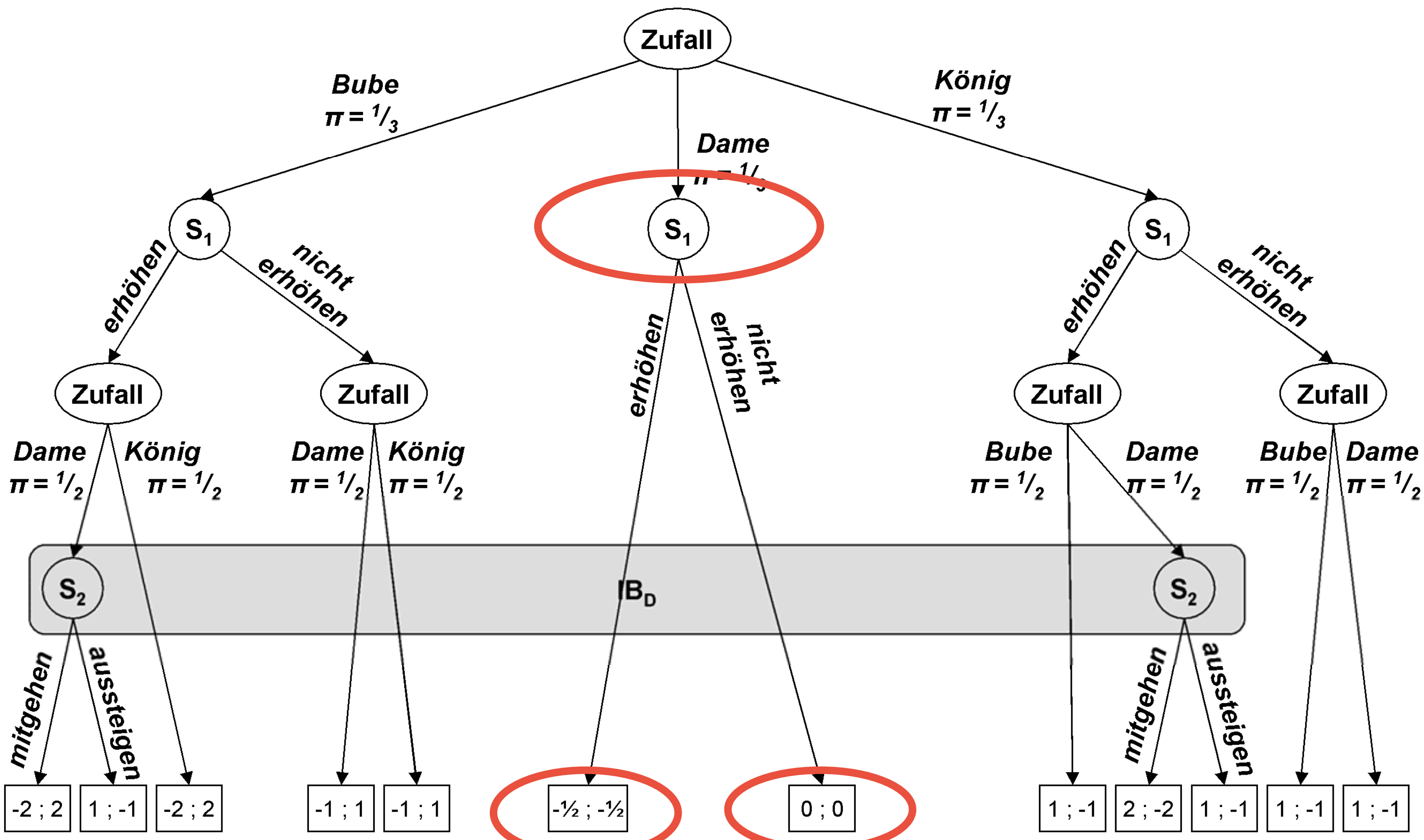


Spieltheorie | Das Pokerspiel | Verbesserte Lösungen | Die Poker Challenge

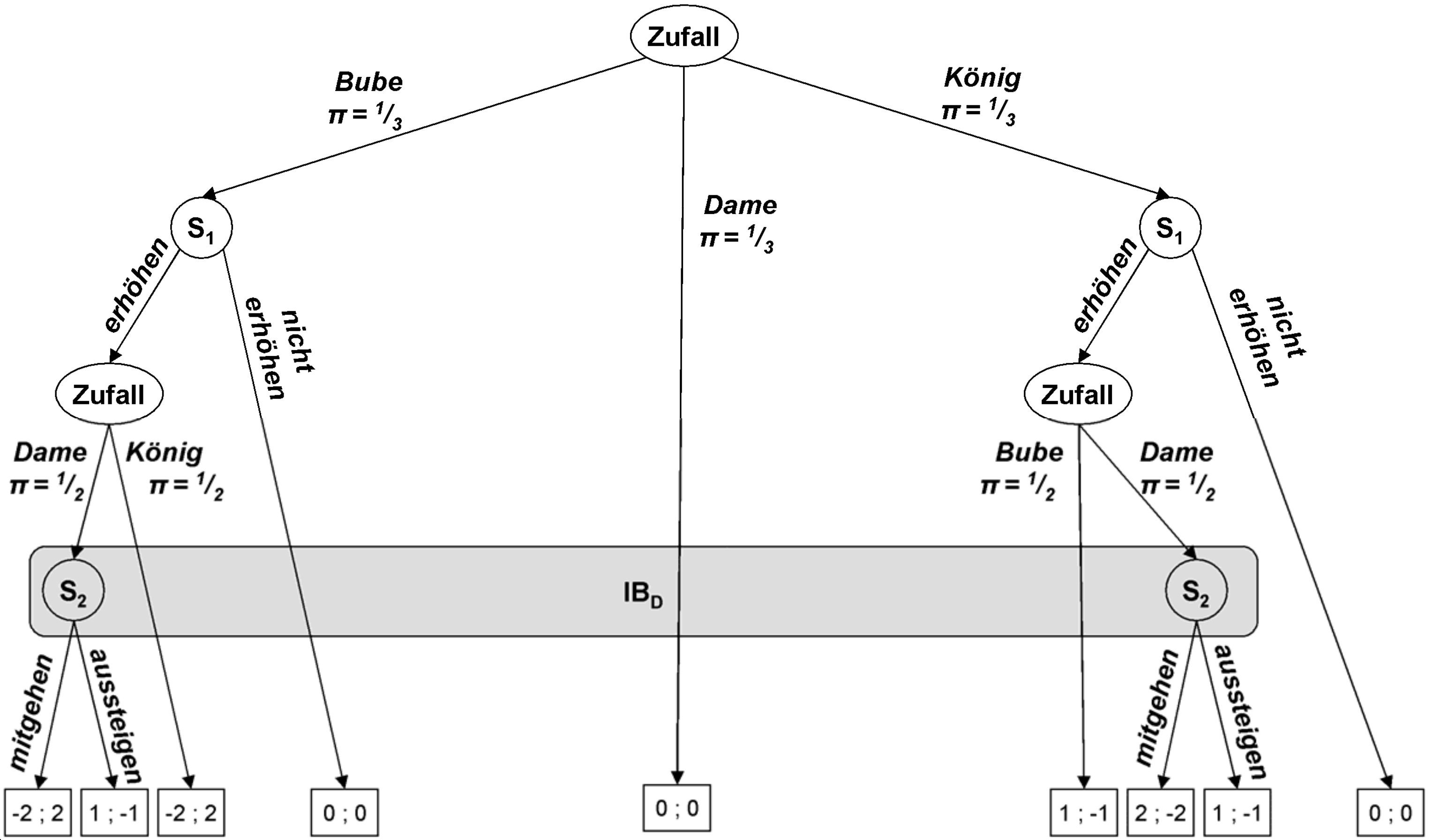
Rückwärtsinduktion und dominierte Strategien → Beispiel



Rückwärtsinduktion und dominierte Strategien → Beispiel



Rückwärtsinduktion und dominierte Strategien → Beispiel



Transformation in Normalform

- Für MaxiMin-Algorithmus: Transformation in Normalform notwendig

- Dazu Bilden der Strategienmenge: Alle möglichen Kombinationen
  - Das sind viele: $|\text{Alternativen}|^{|\text{Entscheidungen}|}$

- Dann: Berechnen der Erwartungswerte für Auszahlungen

Transformation in NF: Beispiel

- Am Beispiel Simples Poker:
  - Strategien: 2^2 (Spieler 1) und 2^1 (Spieler 2)
  - Erwartungswert Auszahlung:

$$E(u_1(B_e, K_e; D_m)) = \frac{1}{6} \cdot -2 + \frac{1}{6} \cdot -2 + \frac{1}{3} \cdot 0 + \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 = -\frac{1}{6}$$

  - Ergebnis:

		Spieler 2	
		D→mitgehen	D→aussteigen
Spieler 1	B→erhöhen; K→erhöhen	$-\frac{1}{6}$	$\frac{1}{6}$
	B→erhöhen; K→lassen	$-\frac{2}{3}$	$-\frac{1}{6}$
	B→lassen; K→erhöhen	$\frac{1}{2}$	$\frac{1}{3}$
	B→lassen; K→lassen	0	0

  - Beachte: Vollständige Form hätte 8 x 8 = 64 Einträge!

Verbesserte Lösungsstrategien

Verbesserte Lösungsstrategien

- spezielle Ansätze

- Optimierung & Approximierung

- Genauigkeitsverlust

Optimierung

- Regelbasierte Darstellung
  - Wie ein Mensch
  - Zwei Kategorien
  - Bei dem Beispiel **Gala**:
    - *chosse(Player, Template, Constraint)*, wobei ein Anwendungsbeispiel
    - *choose(peter, InitialBet, between(0, $money(peter), Bet))*

Optimierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Regelbasierte Darstellung
  - Wie ein Mensch
  - Zwei Kategorien
  - Bei dem Beispiel **Gala**:

Game language
implementiert in Prolog
reale Spielsituationen analysieren

  - *chosse(Player, Template, Constraint)*, wobei ein Anwendungsbeispiel
  - *choose(peter, InitialBet, between(0, $money(peter), Bet))*

Optimierung

- Sequentielle Form
  - Lineare Berechnung möglich
  - von Koller, Megiddo und von Stengel
  - Realisationswert

Spieler $\boxed{k}$

Knoten: $\boxed{p}$

Weg von der Wurzel zur $\boxed{p}$: $\boxed{\delta^k(p)}$

Wahrscheinlichkeit mit $\boxed{\mu_k}$ $\boxed{\delta_k}$ spielen zu können

Optimierung

- Sequentielle Form
  - Lineare Berechnung möglich
  - von Koller, Megiddo und von Stengel
  - Realisationswert

Spieler k
Knoten: p
Weg von der Wurzel zur p : $\delta^k(p)$
Wahrscheinlichkeit mit μ_k δ_k spielen zu können

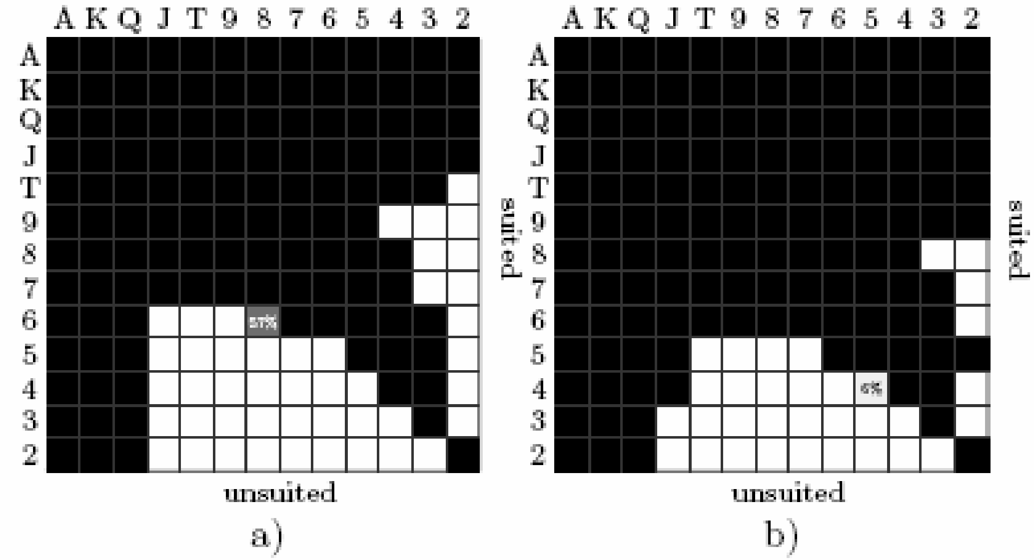
Realisations Wert : $\mu_k(\delta_k)$

Approximation

- Näherung ans Optimum

- Jam/Fold ist fast optimal
  - Für No-limit Texas Hold‘em Turnier
  - Nur preflop wird ausgewertet
  - SB 300, BB 600

- Handlungsraum einschränken

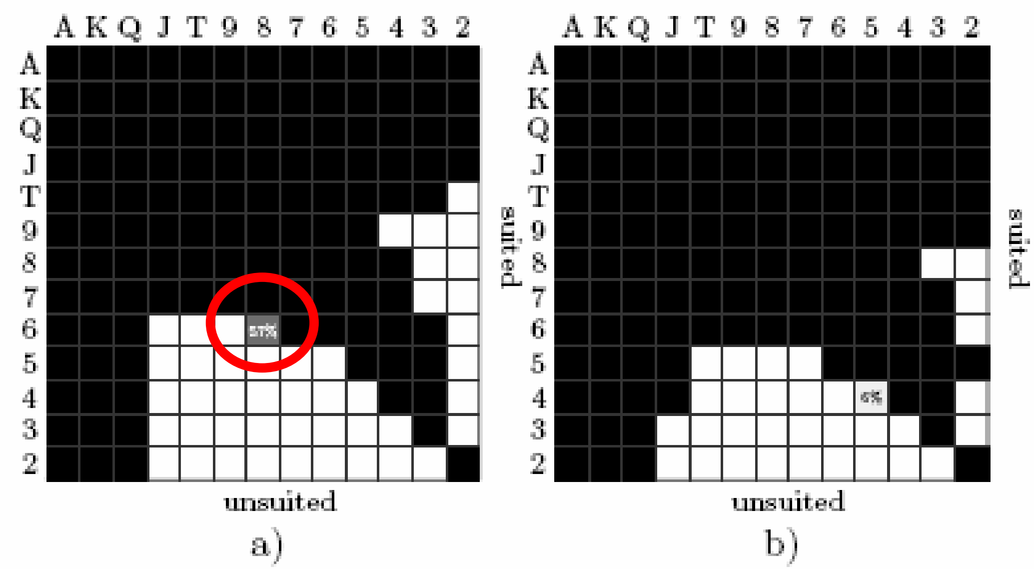


Approximation

- Näherung ans Optimum

- Jam/Fold ist fast optimal
  - Für No-limit Texas Hold‘em Turnier
  - Nur preflop wird ausgewertet
  - SB 300, BB 600

- Handlungsraum einschränken

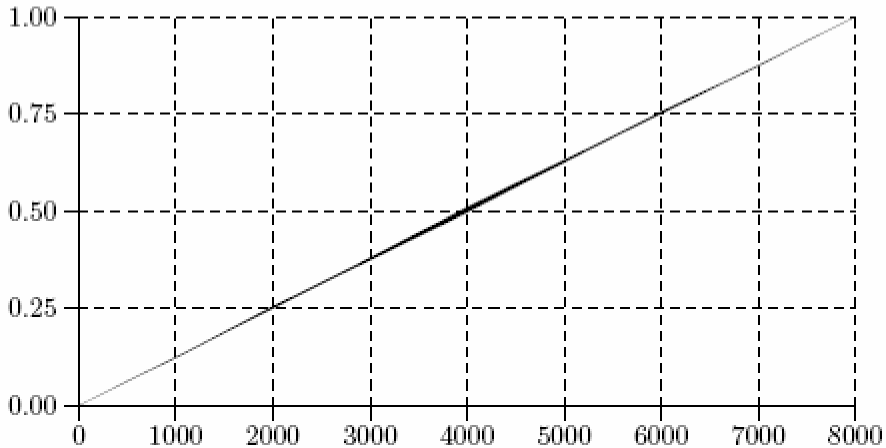


Approximation

- Knotengruppierung
  - Jam/Fold beachtet nur Starthände
  - Die Aktionen der Gegner werden kategorisiert
  - Die Stackgröße ist wichtig
    - Der Stack wird in Intervalle gefasst

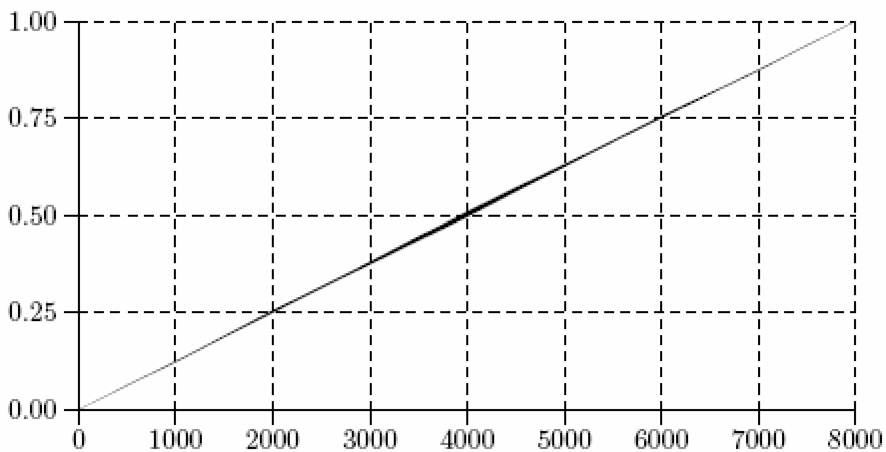
Schranken

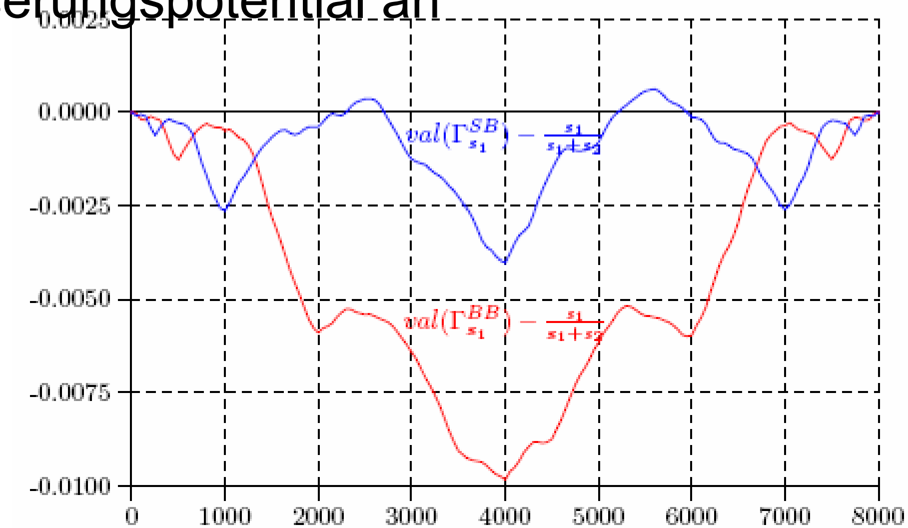
- Es gibt immer eine optimale Strategie
- Man kann die Abweichung berechnen
- Die obere Schranke gibt das Verbesserungspotential an



Schranken

- Es gibt immer eine optimale Strategie
- Man kann die Abweichung berechnen
- Die obere Schranke gibt das Verbesserungspotential an





Genauigkeitsverlust

- Sklansky hat „The System erstellt“
  - Verbesserung zum „revised system“ 5,9%
  - Vergleich mit always jam
- Jam/Fold liegt nur 1,4% von der optimalen Strategie

Umsetzung für unsere Poker Challenge

Umsetzung für unsere Poker Challenge

- Multi-Table Limit Texas Hold'em Turnier
  - Unterschiedlich zu bisherigen Beispielen
  - Dennoch viele Verbesserungen anwendbar

Umsetzung für unsere Poker Challenge

- Knotenanzahl
  - Es gibt viele Faktoren
  - Ohne Approximationen:

$$169! - 162! \sim 3{,}4 * 10^{15}$$

  - D. Billings (Polaris): Ein vollständiges Poker zu berechnen würde einen Spielbaum mit etwa 10 hoch 18 Knoten Aufspannen

Operationalisierung

- Lineares Wachstum erwünscht

- Sequentielle Form
  - allgemein genug
  - Einfache Bildung von LGS

- Regelbasierte
  - Framework nötig
  - Programmierung innerhalb Frameworks sehr effektiv
  - Gala vs. Java

- Schrankenberechnung

Operationalisierung

- Knotenmenge eingrenzen
  - Ausnutzung von Symetrien
  - Kategorisierung der Aktionen
  - Größe des eigenen Stacks
  - Verteilung des Stacks
  - Viele Weitere

Abschluss

- Grundlagen der Spieltheorie wichtig für die Berechnung
- Optimierung
- Approximation
- → Ziel: Rechenaufwand sparen

- Vielen Dank für die Aufmerksamkeit