Absprachen-resistente Kostenaufteilungs-Mechanismen

Design-Techniken, Analysen, Abwägungen

Florian Schoppmann, Promotionsprüfung, 15. Juni 2009

Motivation: Gemeinsame Projekte

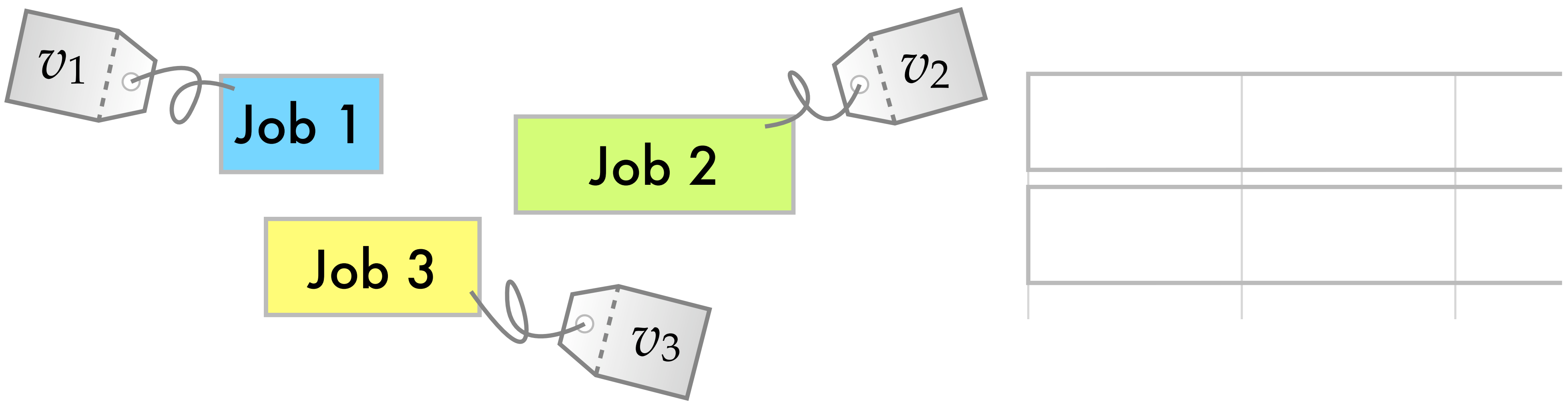
"**Wer** sollte an einem gemeinsamen Projekt teilnehmen und **zu welchem Preis**?"

Automatisierte Verhandlungen in der Logistik

Infrastruktur für Breitband-Internetzugänge schaffen

Car Sharing

Aufnahme in einen Ablaufplan



- Wer sollte teilnehmen und zu welchem Preis?
  - Menge von Jobs $Q \subseteq \{1, \ldots, n\}$
  - Konstenanteile $x_i \in [0, v_i]$
  - Zulässiger Ablaufplan mit Länge C

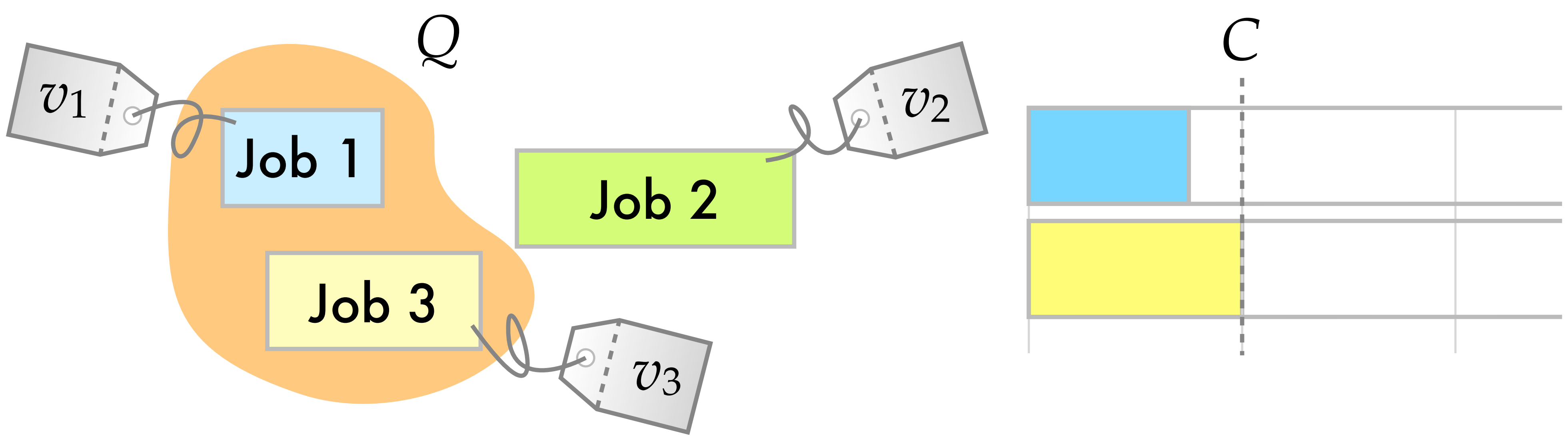
α-Effizienz ($\alpha \geq 1$)

- $C + \sum_{i \notin Q} v_i$ ist α-approximativ

β-Budget-Balance ($\beta \geq 1$)

- $C \leq \sum x_i \leq \beta \cdot OPT(Q)$

Aufnahme in einen Ablaufplan



- Wer sollte teilnehmen und zu welchem Preis?
  - Menge von Jobs $Q \subseteq \{1, \ldots, n\}$
  - Konstenanteile $x_i \in [0, v_i]$
  - Zulässiger Ablaufplan mit Länge C

α-Effizienz ($\alpha \geq 1$)

- $C + \sum_{i \notin Q} v_i$ ist α-approximativ

β-Budget-Balance ($\beta \geq 1$)

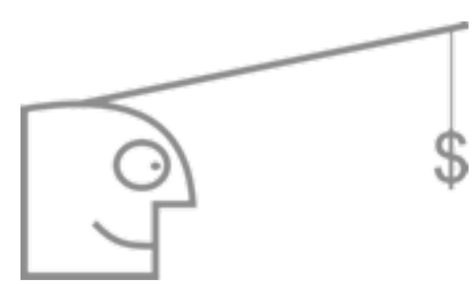
- $C \leq \sum x_i \leq \beta \cdot OPT(Q)$

Mechanismen-Design

- Wertschätzung v_i ist nur Spieler i selbst bekannt
- Spieler agieren strategisch, können $b_i \neq v_i$ bieten
  - Optimieren eigenen Netto-Nutzen $= \begin{cases} v_i - x_i & \text{wenn } i \in Q \\ 0 & \text{sonst} \end{cases}$
  - Kennen den Mechanismus

$\boldsymbol{b} \longrightarrow$  $\longrightarrow Q$, $\longrightarrow \boldsymbol{x} = (x_1, \ldots, x_n)$

- Wichtige Anforderung: Anreiz-Kompatibilität

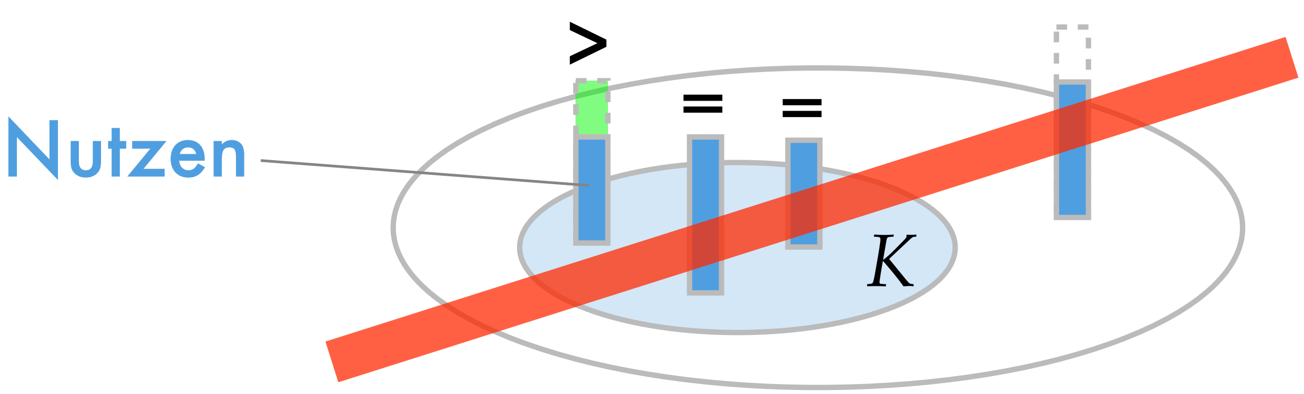


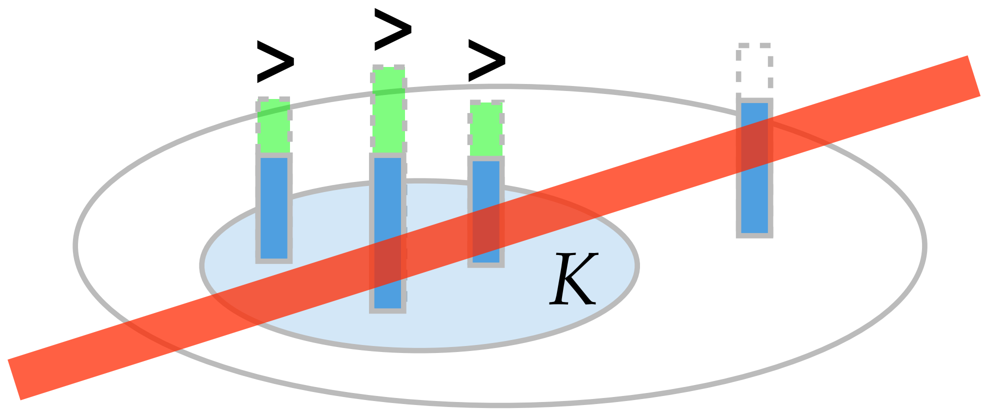
Absprachen-Resistenz

- Angenommen, eine Koalition K weicht von v ab
  - Gruppen-strategisch robust (GSP, group-strategyproof):

    Ein Mitglied aus K gewinnt $\Rightarrow$ anderes Mitglied verliert
  - Schwach Gruppen-strategisch robust (WGSP, weak GSP):

    Nicht alle Mitglieder in K gewinnen



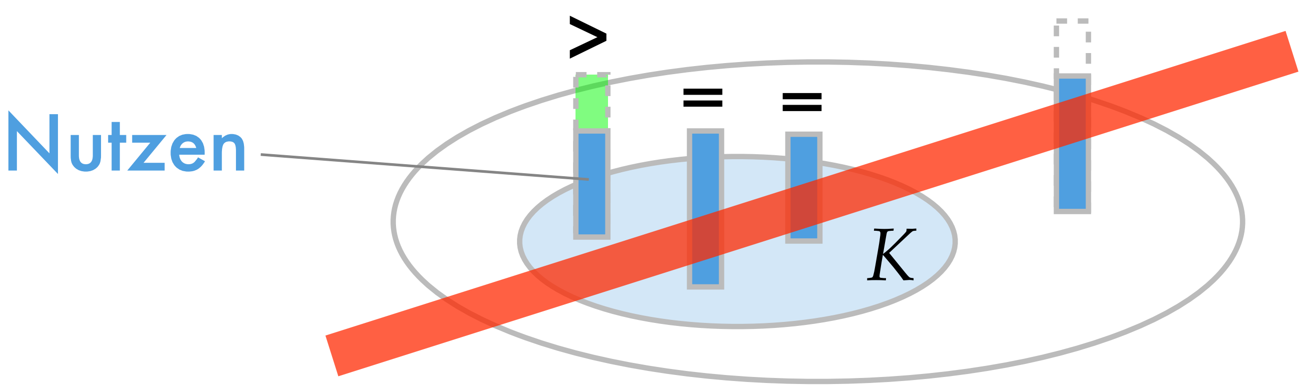


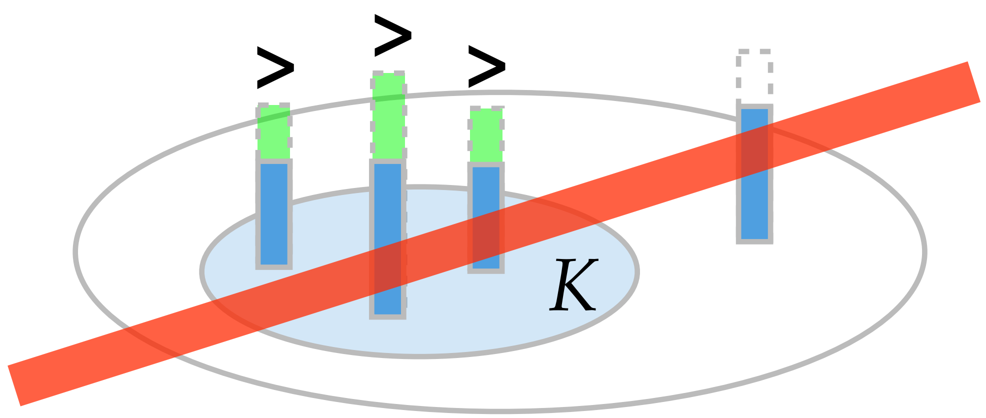
Absprachen-Resistenz

- Angenommen, eine Koalition K weicht von v ab
  - Gruppen-strategisch robust (GSP, group-strategyproof):

    Ein Mitglied aus K gewinnt $\Rightarrow$ anderes Mitglied verliert
  - Schwach Gruppen-strategisch robust (WGSP, weak GSP):

    Nicht alle Mitglieder in K gewinnen

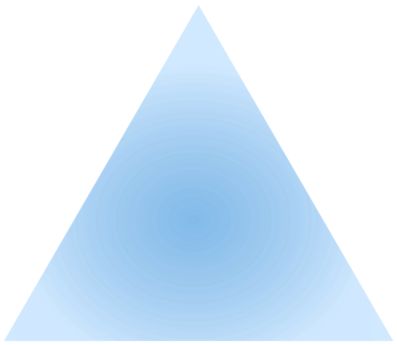




- Beschränkt auf $|K| = 1$: Strategisch robust (SP)

Kostenaufteilung

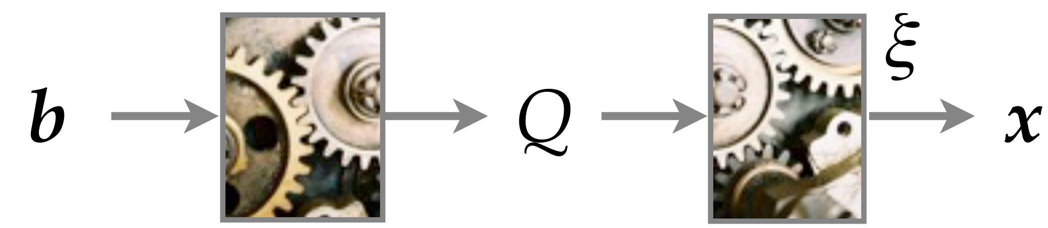
- Fundamentales Problem
  - Kooperative Spiele ↔ Approximations-Algorithmen
- Unumgängliche Abwägungen?



- Eines der ersten Probleme der algorithmischen Spieltheorie
  - Feigenbaum et al. (STOC'00), Jain & Vazirani (STOC'01)

Bisherige Techniken

- Separabilität: $\boldsymbol{x} = \xi(Q)$



- GSP: Moulin-Mechanismen (Moulin, Soc Choice Welf'99)
  - Cross-monotone Kostenanteile: $\xi_i(S \cup j) \leq \xi_i(S)$
  - Wähle größte $\boldsymbol{b}$-zulässige Menge, d.h. $\forall i \in Q$: $b_i \geq \xi_i(Q)$

$Q := \{1, \ldots, n\}$;

solange $\exists\, i$: $b_i < \xi_i(Q)$

$\quad Q := Q \setminus i$

- WGSP: Azyklische Mechanismen (Mehta et al., EC'07)

Meine Dissertation

① Lexikographische Maximierung (Kapitel 3, MFCS'07)

- Erste alternative Design-Technik für GSP Mechanismen

Meine Dissertation

① Lexikographische Maximierung (Kapitel 3, MFCS'07)

- Erste alternative Design-Technik für GSP Mechanismen

② Kostenaufteilung ohne Indifferenzen (Kapitel 4, WINE'07)

- Verbesserte Approximation-Garantien durch leicht abgeschwächte Absprachen-Resistenz

Meine Dissertation

① Lexikographische Maximierung (Kapitel 3, MFCS'07)

- Erste alternative Design-Technik für GSP Mechanismen

② Kostenaufteilung ohne Indifferenzen (Kapitel 4, WINE'07)

- Verbesserte Approximation-Garantien durch leicht abgeschwächte Absprachen-Resistenz

③ Spielt Koalitionsgröße eine Rolle? (Kapitel 5, WINE'08)

- Im Fall von GSP: "Nein"

Meine Dissertation

① Lexikographische Maximierung (Kapitel 3, MFCS'07)

- Erste alternative Design-Technik für GSP Mechanismen

② Kostenaufteilung ohne Indifferenzen (Kapitel 4, WINE'07)

- Verbesserte Approximation-Garantien durch leicht abgeschwächte Absprachen-Resistenz

③ Spielt Koalitionsgröße eine Rolle? (Kapitel 5, WINE'08)

- Im Fall von GSP: "Nein"

④ Unterschiedlichen Teilnahme-Stufen (Kapitel 6, SAGT'08)

- Erste Design-Technik für GSP Mechanismen

① Lexikographische Maximierung

- Moulin: Q maximiert Netto-Nutzen für alle

① Lexikographische Maximierung

- Moulin: Q maximiert Netto-Nutzen für alle
  - Was können wir ohne Cross-Monotonie tun?
  - Beispiel: Beide Spieler bieten 2

Q	{1}	{2}	{1, 2}
x_1, x_2	1, 0	0, 1	2, 1
Nutzen	1 0	0 1	0 1

① Lexikographische Maximierung

- Moulin: Q maximiert Netto-Nutzen für alle
  - Was können wir ohne Cross-Monotonie tun?
  - Beispiel: Beide Spieler bieten 2

Q	{1}	{2}	{1, 2}
x_1, x_2	1, 0	0, 1	2, 1
Nutzen	1, 0	0, 1	0, 1

- Idee: Lexikographische Maximierung des Nutzen-Vektors

Symmetrische Mechanismen

- Hier einfachster Spezialfall
- Nur zwei Preise: $h > l$
  - Spieler mit hoher Nummer zahlen l
- Anzahl Spieler, die l zahlen:
  - hängt nur von $|Q|$ ab
  - monoton in $|Q|$

$\|Q\|$	Preise					
1	h					
2	l	l				
3	h	l	l			
4	h	h	l	l		
5	l	l	l	l	l	l

Symmetrische Mechanismen

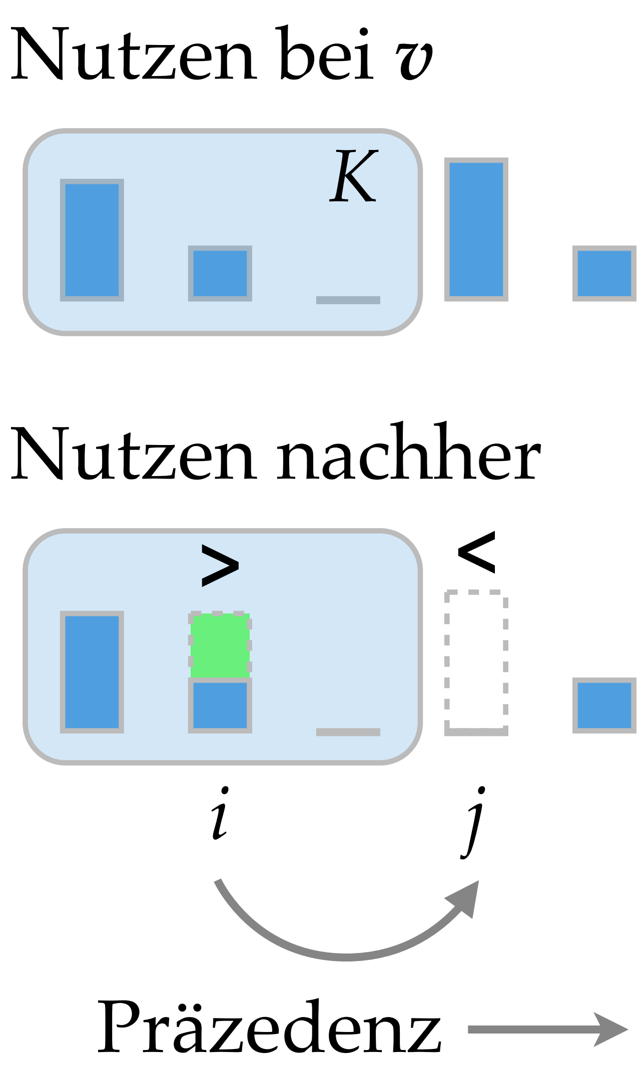
- Hier einfachster Spezialfall
- Nur zwei Preise: $h > l$
  - Spieler mit hoher Nummer zahlen l
- Anzahl Spieler, die l zahlen:
  - hängt nur von $|Q|$ ab
  - monoton in $|Q|$
- Beispiel: $Q = \{ 1, 4, 5 \}$

$\|Q\|$	Preise					
1	h					
2	l	l				
3	h	l	l			
4	h	h	l	l		
5	l	l	l	l	l	l

i	1	2	3	4	5
x_i	h	0	0	l	l

Beweis: Gruppen-strategische Robustheit

- Annahme: Erfolgreiche Koalition K
  - Sei $(Q, \boldsymbol{x})$ bisheriger Ausgang für $\boldsymbol{v}$ und $(Q', \boldsymbol{x}')$ Ausgang nach Manipulation durch K
  - Sowohl Q als auch Q' sind $\boldsymbol{v}$-zulässig
- Ein Spieler i verbessert Nutzen
  - $i \in Q'$ und $x'_i = l$
  - Ein $j > i$ verliert Nutzen; deshalb $j \notin Q'$
- Q' hat nicht lexikographisch maximalen Nutzenverktor!



Ergebnisse zu ①

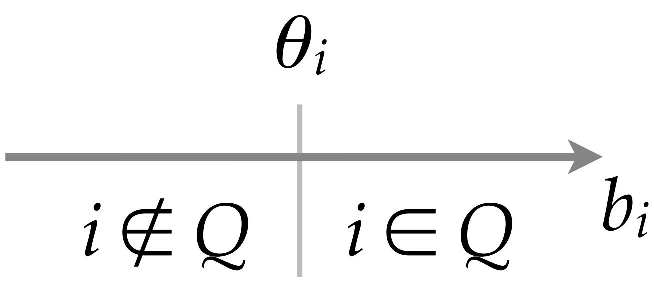
- Symmetrische Mechanismen
  - Erste alternative Technik für GSP
  - $\frac{\sqrt{17}+1}{4} \approx 1{,}28$-Budget-Balance für subadditive symmetrische Kosten, d.h. $C(A \cup B) \leq C(A) + C(B)$ und $C(A) = c(|A|)$
  - Untere Schranke für Moulin-Mechanismen ist hier 2-Budget-Balance

$\|Q\|$	Preise					
1	2					
2	1	1				
3	5	1	1			
4	5	4	1	1		
5	3	1	1	1	1	
6	1	1	1	1	1	1

- Falls $n > 3$, so sind 1-Budget-Balance und GSP i.A. unmöglich, selbst für symmetrische Kosten

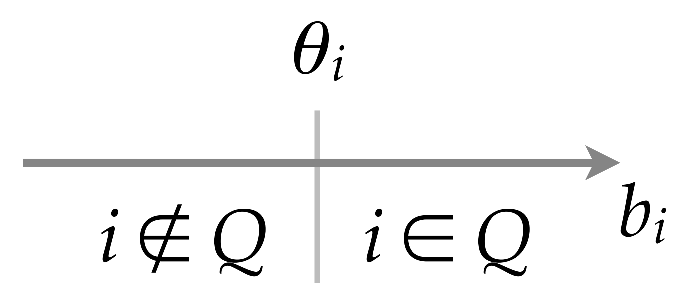
② Kostenaufteilung ohne Indifferenzen

- Schwellwert-Eigenschaft



② Kostenaufteilung ohne Indifferenzen

- Schwellwert-Eigenschaft



- Indifferente Spieler sind problematisch
  - Stabile Koalition, wenn Spieler Service verliert?
- Absprachen-Resistenz zwischen GSP und WGSP
  - Schwach Gruppen-strategisch robust gegen Service-interessierte Spieler (WSGSP): Ein Koalitions-Mitglied gewinnt $\Rightarrow$ anderes Mitglied verliert Nutzen oder Service

Egalitäre Mechanismen

$Q := \emptyset$

Solange Spieler $\notin Q$ vorhanden

$S :=$ Kosten-effizienteste Teilmenge der Spieler $\notin Q$,
d.h. $a := \frac{C(Q \cup S) - C(Q)}{|S|}$ ist minimal

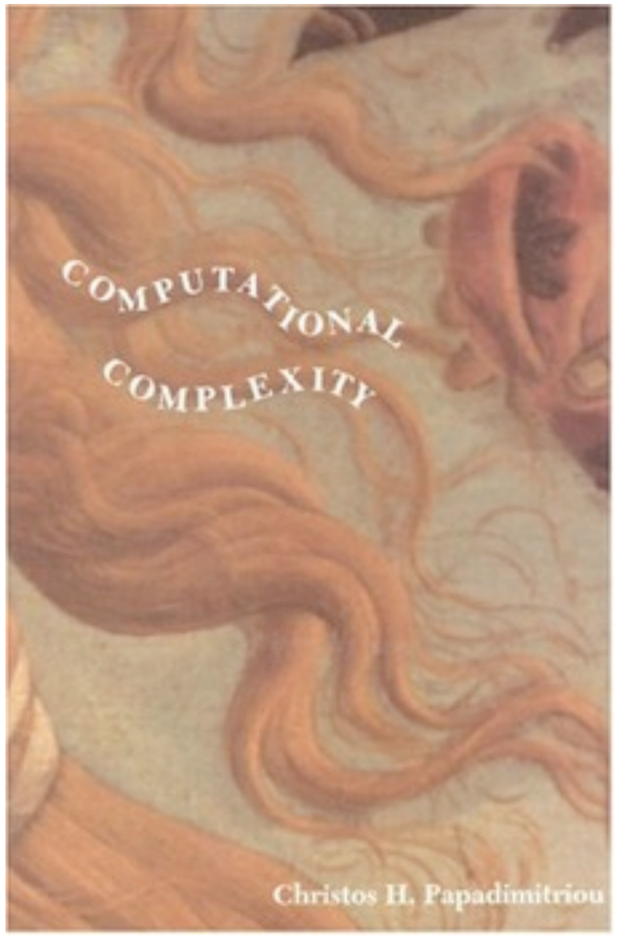
Lösche alle Spieler mit Gebot $< a$

Wenn niemand gelöscht **dann** $Q := Q \cup S$ und $x_i := a$ für alle $i \in S$

- Strategisch robust wegen nicht-fallender Preise
- Für WSGSP mehr Aufwand nötig

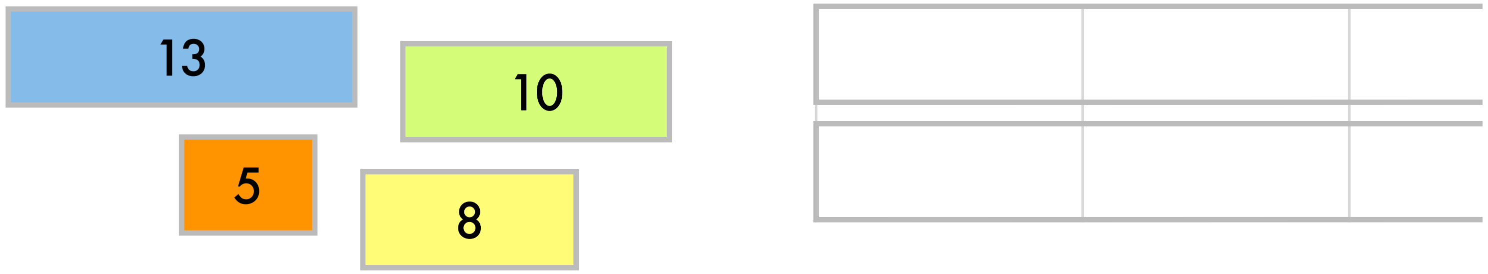
Effiziente Berechenbarkeit

- Finden der Kosten-effizientesten Menge problematisch
  - Kosten stammen oft von Lösungen NP-schwerer Probleme
  - Testen aller 2^n Teilmengen zu aufwändig
- Idee: Monotone Algorithmen



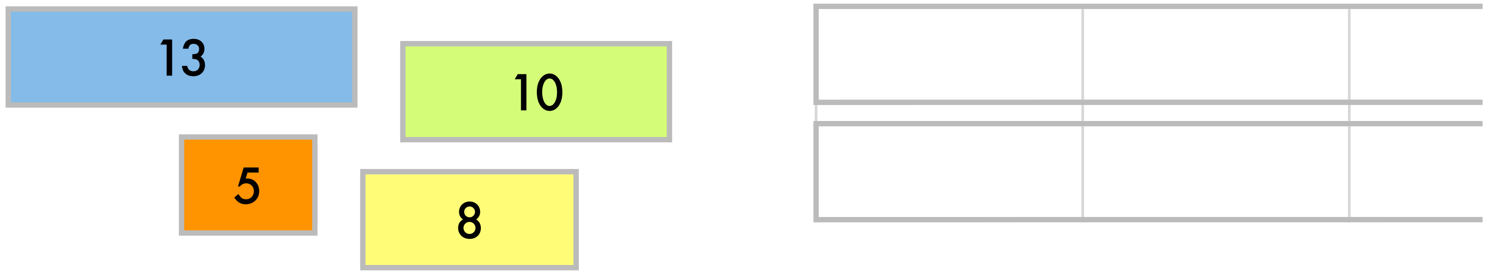
Kosten-effizienteste Menge finden

- Manche Probleme sind „inhärent monoton“



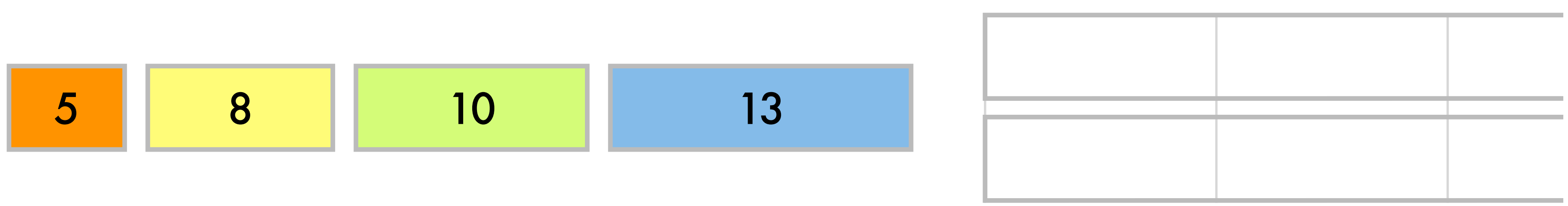
Kosten-effizienteste Menge finden

- Manche Probleme sind „inhärent monoton“
- Minimaler Quotient $\frac{C(S)}{|S|}$ einfach zu bestimmen
  - Für jede mögliche Anzahl Jobs teste nur jeweils kleinste Jobs



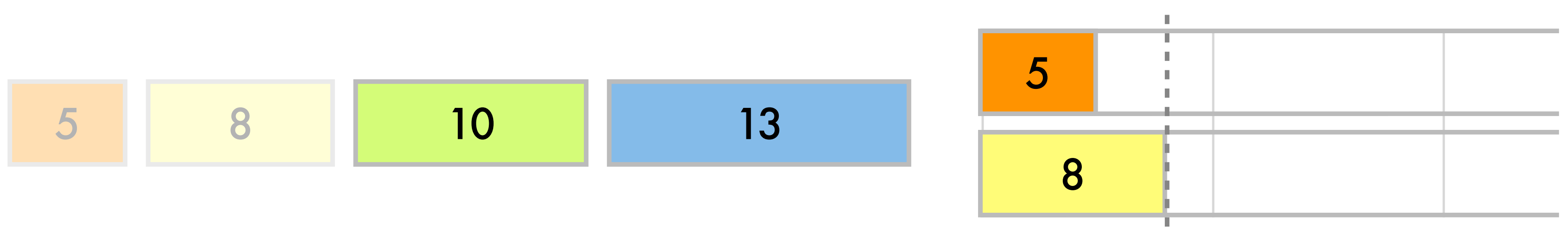
Kosten-effizienteste Menge finden

- Manche Probleme sind „inhärent monoton“
- Minimaler Quotient $\frac{C(S)}{|S|}$ einfach zu bestimmen
  - Für jede mögliche Anzahl Jobs teste nur jeweils kleinste Jobs



Kosten-effizienteste Menge finden

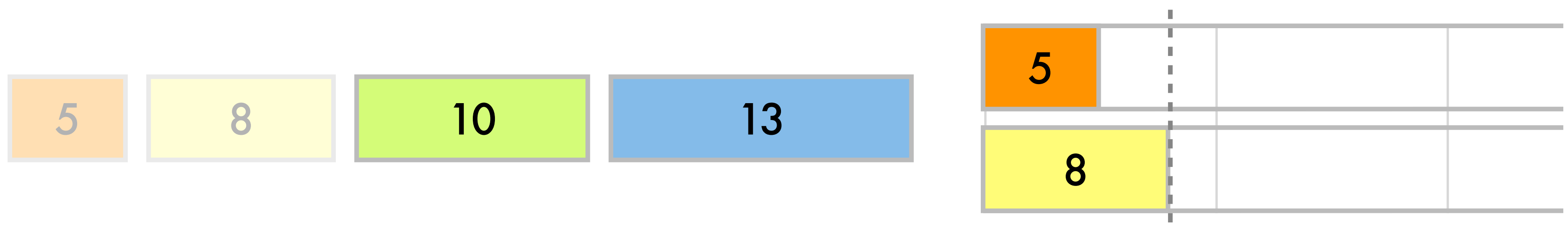
- Manche Probleme sind „inhärent monoton“
- Minimaler Quotient $\frac{C(S)}{|S|}$ einfach zu bestimmen
  - Für jede mögliche Anzahl Jobs teste nur jeweils kleinste Jobs



$$\frac{C(S)}{|S|} = 4$$

Kosten-effizienteste Menge finden

- Manche Probleme sind „inhärent monoton“
- Minimaler Quotient $\frac{C(S)}{|S|}$ einfach zu bestimmen
  - Für jede mögliche Anzahl Jobs teste nur jeweils kleinste Jobs



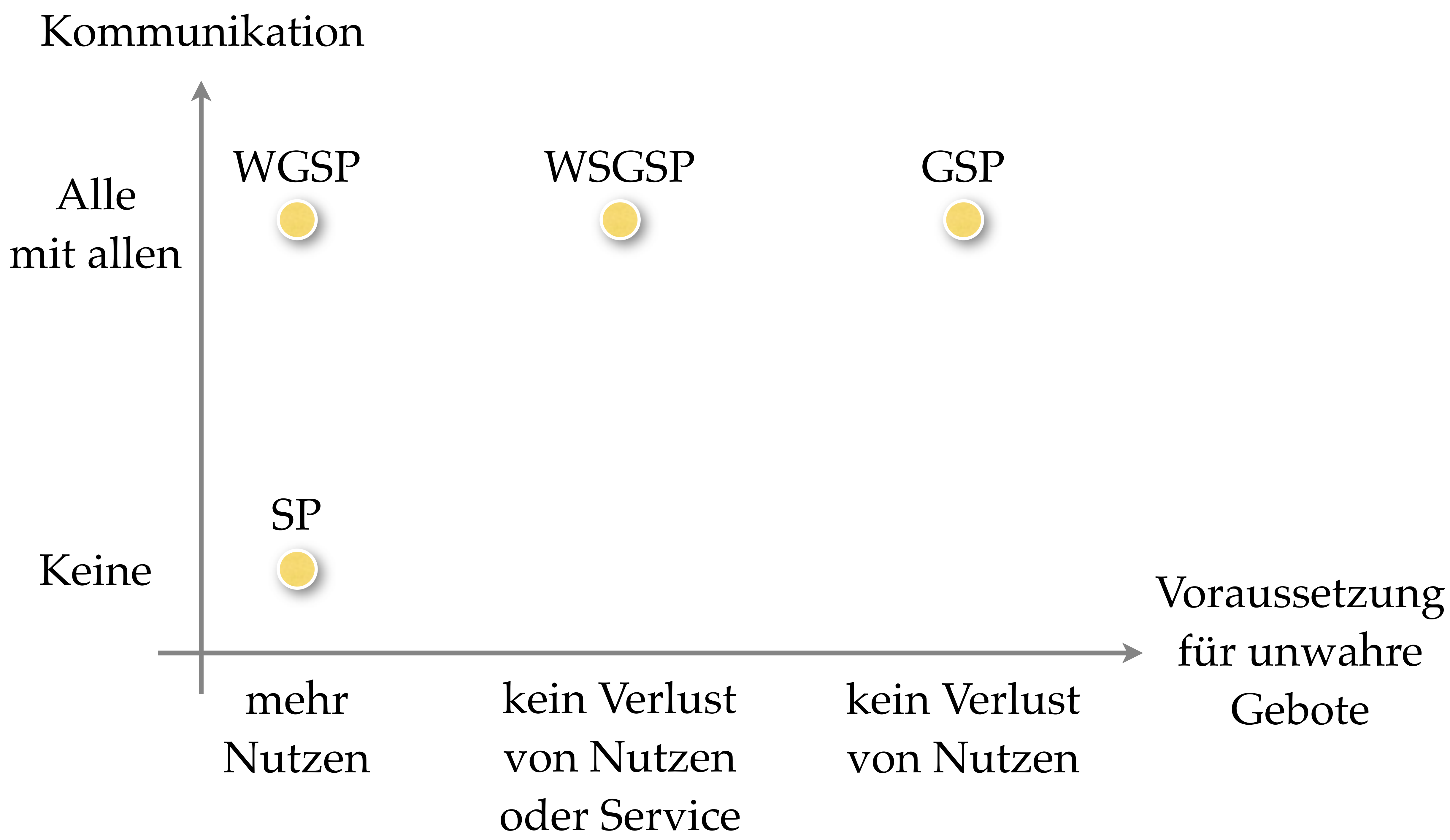
$$\frac{C(S)}{|S|} = 4$$

- Monotone Approximations-Algorithmen?
  - Entwurfsmuster: Runden und optimal lösen
  - modifiziertes PTAS von Hochbaum/Shmoys, SICOMP’88

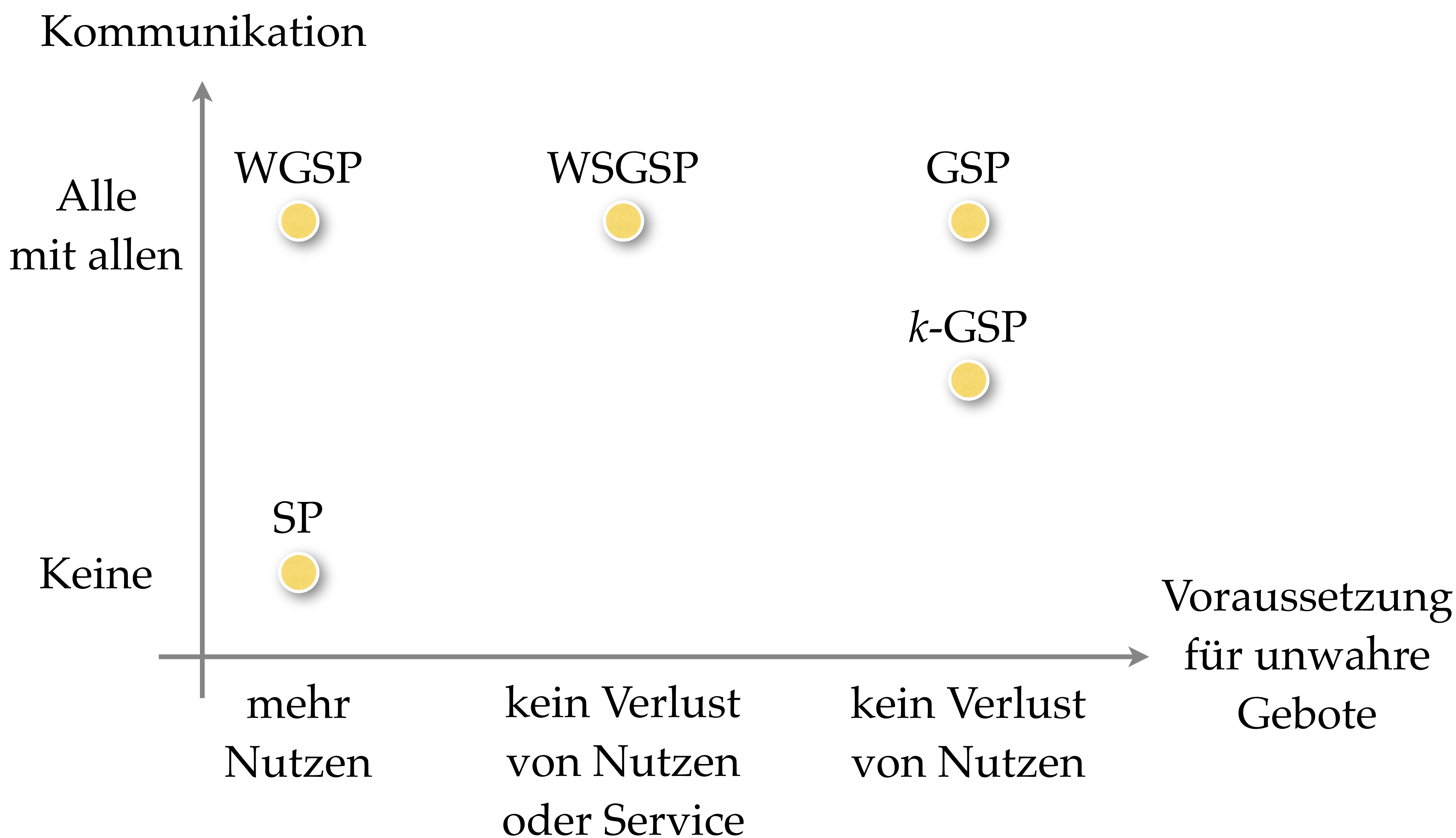
Ergebnisse zu ②

- Egalitäre Mechanismen sind 1-Budget-balanciert, für subadditive Kosten auch $2H_n$-effizient
  - Asymptotisch optimal – selbst für Mechanismen, die nur SP sind (Dobzinski et al., SAGT'08)
- Gerüst für Polynomialzeit-Berechnung
- Azyklische Mechanismen sind WSGSP, und egalitäre Mechanismen sind azyklische Mechanismen

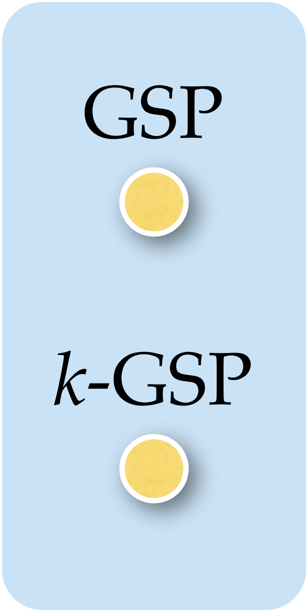
③ Spielt Koalitionsgröße eine Rolle?



③ Spielt Koalitionsgröße eine Rolle?



③ Spielt Koalitionsgröße eine Rolle?



- Hauptergebnis: GSP ⇔ 2-GSP + Separabel
  - (“GSP ⇒ Separabel” von Moulin, Soc Choice Welf’99)

Design-Techniken, Analysen, Abwägungen

- Beispiel Polynomzeit-Mechanismen für $Q \,||\, C_{\max}$

Mechanismen	Moulin	① Symmetrische	② Egalitäre
Nutzen-Optimierung	Für alle	Rangfolge	Für alle, wenn möglich
Absprachen-Resistenz	GSP	GSP	WSGSP
Budget-Balance	$2\,d$	$1{,}28 \cdot d$	2
Ökonomische Effizienz	$2\,d \cdot (H_n + 1)$	$\Theta(n)$	$4\,H_n$

d: Anzahl Job-Längen, $H_n = \sum_{i=1}^{n} \frac{1}{i}$: n-te harmonische Zahl

Design-Techniken, Analysen, Abwägungen

Mechanismen	Moulin	① Symmetrische	② Egalitäre
Nutzen-Optimierung	Für alle	Rangfolge	Für alle, wenn möglich
Absprachen-Resistenz	++	++	+
Budget-Balance	o	+	++
Ökonomische Effizienz	+	o	++

Wichtige offene Fragen

- Allgemeinere Methoden zur Verwendung von Approx'algorithmen in Kostenaufteilungs-Mechanismen?
- Wie lassen sich GSP oder WSGSP Mechanismen vollständig charakterisieren?
  - GSP mit guter Budget-Balance und Effizienz?
- WSGSP erlaubt dieselben Approx'garantien wie SP
  - Falls $P \neq NP$: Gilt dies immer noch bei Beschränkung auf Polynomzeit-Mechanismen?

Vielen Dank!

- Azyklische Mechanismen
- 2-GSP
- Monotone Approx' Algorithmen
- Symmetrische Mechanismen
- Übersicht Optimierungsprobleme
- Untere Schranken
- Unterschiedliche Teilnahme-Stufen

Azyklische Mechanismen

(Mehta et al. EC'07)

- $\tau_i(S)$ ist Priorität für Spieler i in S

$Q := \{1, \ldots, n\}$;

solange $\exists\, i$: $b_i < \xi_i(Q)$

Wähle $T \subseteq \{\, \mathrm{i} \in Q \mid b_i < \xi_i(Q)$ und $\tau_i(Q)$ minimal $\}$

$Q := Q \setminus T$

- ξ und τ und sind gültig:
  - $\xi_i(S \setminus T) = \xi_i(S)$ wenn $\tau_k(S) > \tau_i(S)$ für alle $k \in S \cap T$
  - $\xi_i(S \setminus T) \geq \xi_i(S)$ wenn $\tau_k(S) \geq \tau_i(S)$ für alle $k \in S \cap T$

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

1.5	0.5	2	0.8	3	1

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

1.5	0.5	2	0.8	3	1

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

1.5	0.5	2	0.8	3	1

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

1.5	0.5	2	0.8	3	1

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

1.5 0.5 2 0.8 3 1

Symmetrische Mechanismen berechnen

Lösche alle Spieler i mit $b_i < l$

wiederhole

Wenn möglich, nehme alle Spieler i mit $b_i > l$ auf (lösche ggf. indifferente Spieler)

Sonst: Akzeptiere letzten Spieler i für h oder lösche ihn, wenn $b_i < h$

|Q|	Preise					
1	2					
2	1	1				
3	2	1	1			
4	2	2	1	1		
5	2	1	1	1	1	
6	1	1	1	1	1	1

Gebote:

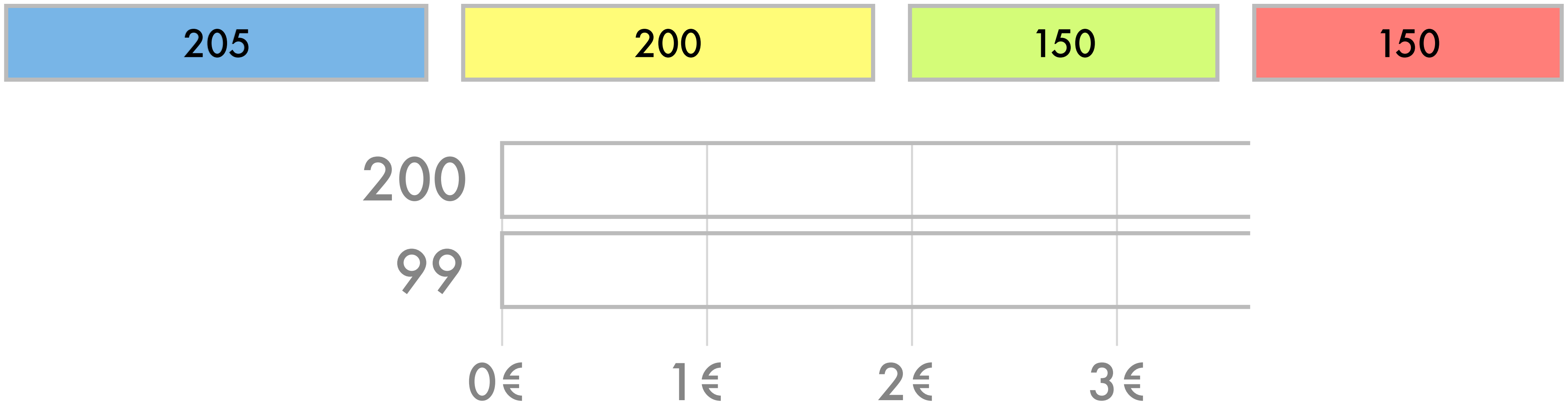
1.5 0.5 2 0.8 3 1

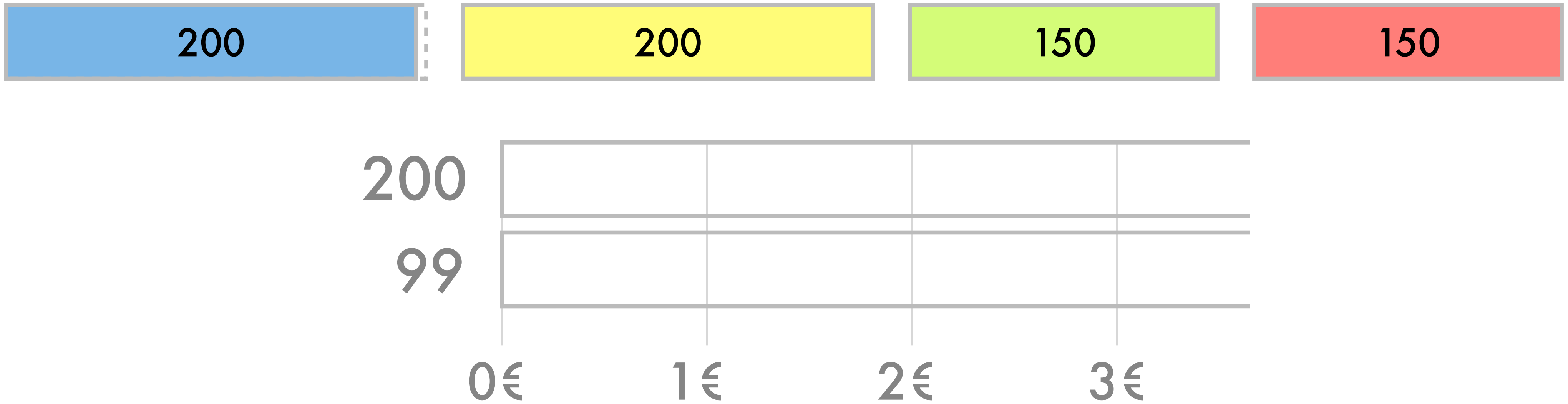
Symmetrische Mechanismen

- $\xi_1(p) \geq \xi_2(p) \geq \ldots$
- $l_1 \geq l_2 \geq \ldots$
- In einem Segment:
  - $l_p = \xi_1(p)$ konstant
  - $\xi_d(p) \leq \xi_d(r)$ wenn $d \leq d_p$ und $d \leq d_r$
  - $d_p \leq d_{p-1} + 1$
  - $d_p \leq \lambda$ wenn $\xi_\lambda(p) \leq \xi_\lambda(p-1)$
- Beispiel: $d_4 = 2$, $l_4 = 1$, und $\xi_2(4) = 4$

	|Q|	Preise					
	1	2					
Segment	2	1	1				
	3	5	1	1			
	4	5	4	1	1		
	5	3	1	1	1	1	
	6	1	1	1	1	1	1

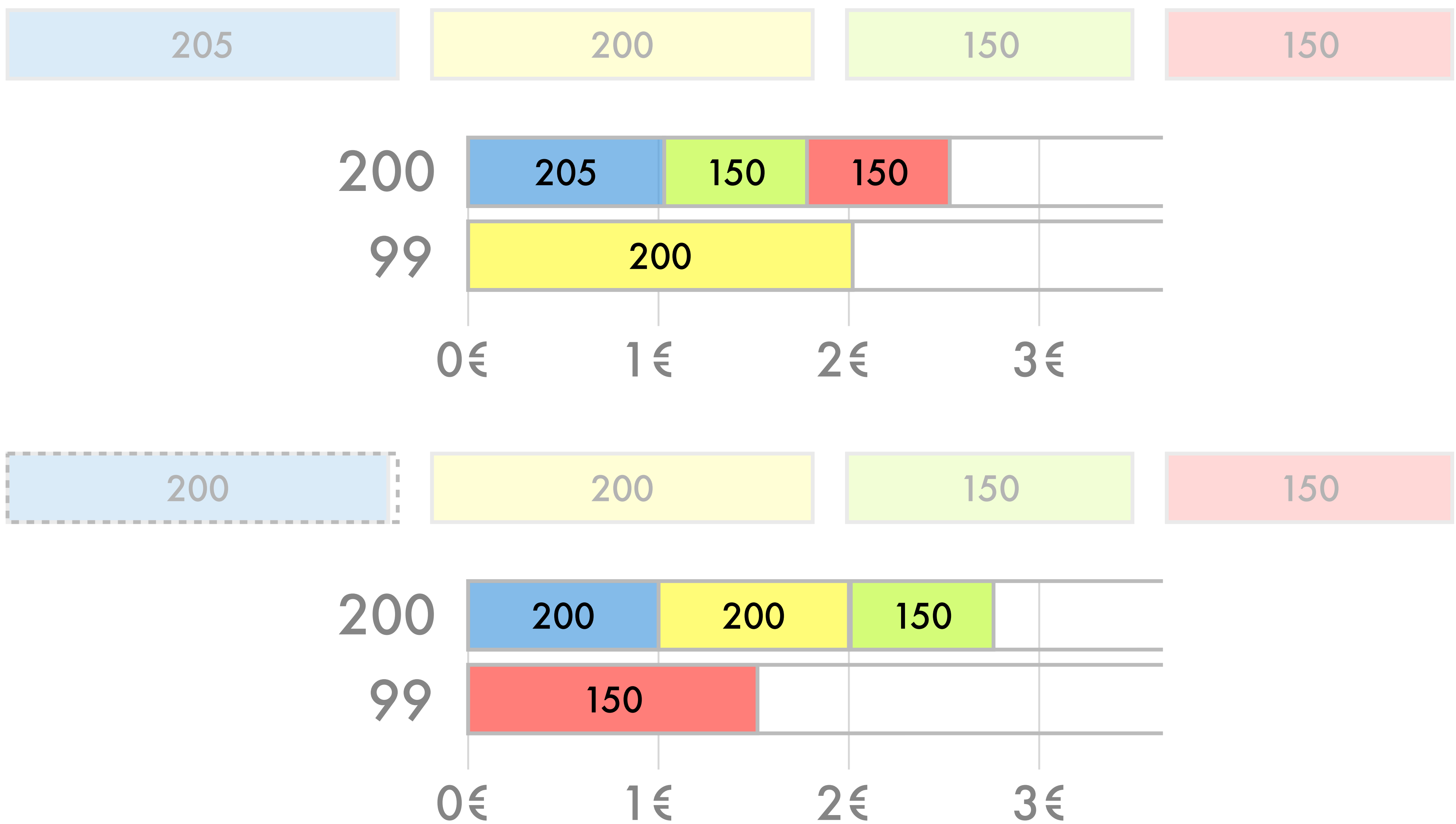
LPT ist nicht monoton





A4

LPT ist nicht monoton



Entwurfsmuster: Runden und optimal lösen

- Monotoner 2-Approx.-Algorithmus für $Q \mid\mid C_{max}$
- Reduktion auf Bin-Packing mit Kapazitäten
  - "First Fit Decreasing" ist optimal (und daher monoton) when Objektgrößen teilbar sind

  - Kapazität = Machinen-Geschwindigkeit · Skalierungsfaktor
  - Binäre Suche nach minimalem Faktor → Gesamtlänge

Entwurfsmuster: Runden und optimal lösen

- Monotoner 2-Approx.-Algorithmus für $Q \,||\, C_{\max}$
- Reduktion auf Bin-Packing mit Kapazitäten
  - "First Fit Decreasing" ist optimal (und daher monoton) when Objektgrößen teilbar sind

  - Kapazität = Machinen-Geschwindigkeit · Skalierungsfaktor
  - Binäre Suche nach minimalem Faktor → Gesamtlänge

Entwurfsmuster: Runden und optimal lösen

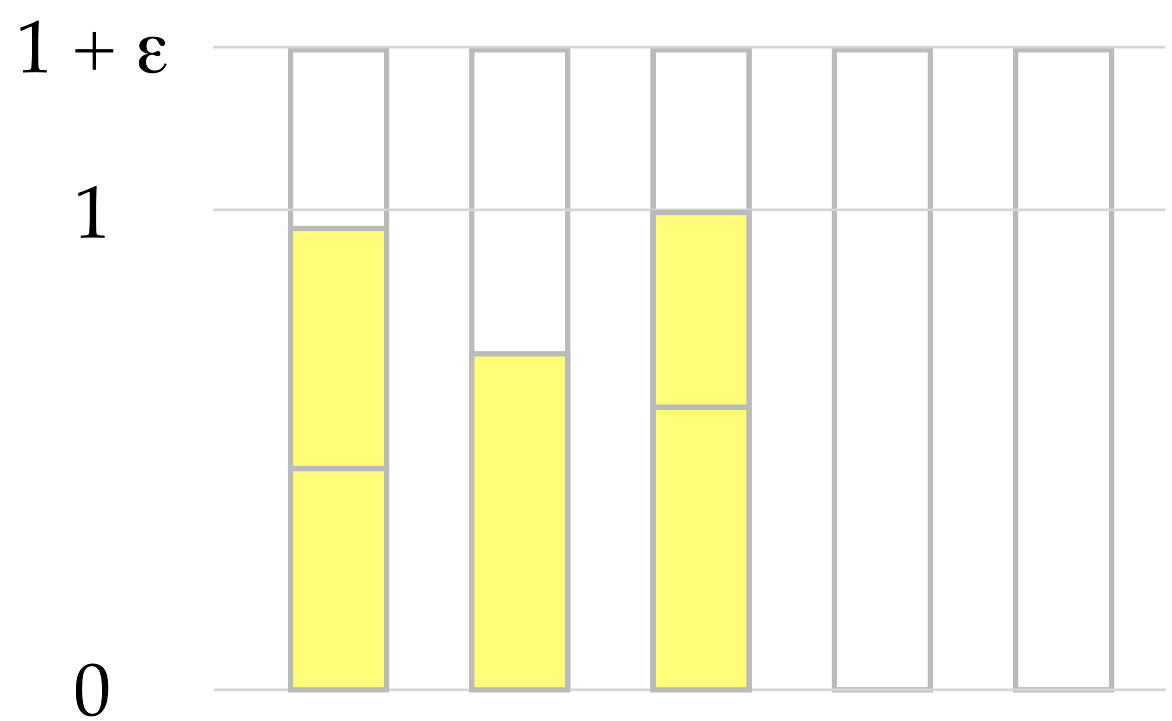
- Monotoner 2-Approx.-Algorithmus für $Q \,||\, C_{max}$
- Reduktion auf Bin-Packing mit Kapazitäten
  - “First Fit Decreasing” ist optimal (und daher monoton) when Objektgrößen teilbar sind

  - Kapazität = Machinen-Geschwindigkeit · Skalierungsfaktor
  - Binäre Suche nach minimalem Faktor → Gesamtlänge

ε-Dualer Approximations-Algorithmus für Bin-Packing

- Eingabe: n Objekte
- Ausgabe: Anzahl benötigter Behälter

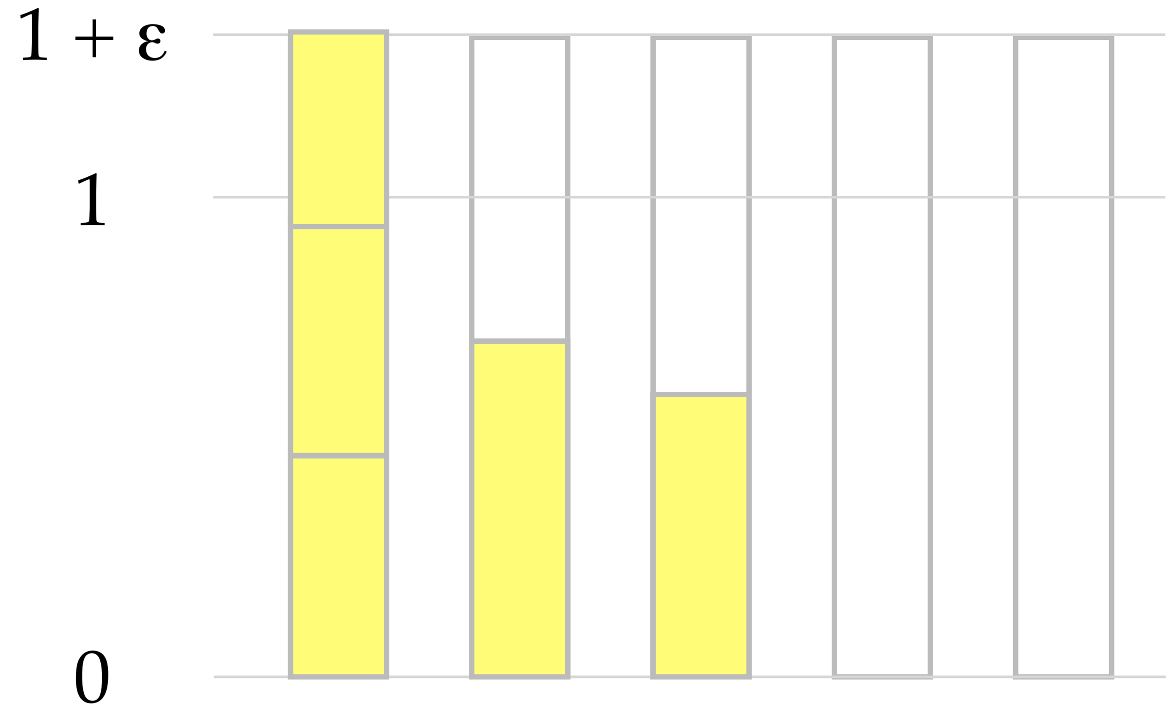
Optimale Aufteilung



ε-Dualer Approximations-Algorithmus für Bin-Packing

- Eingabe: n Objekte
- Ausgabe: Anzahl benötigter Behälter

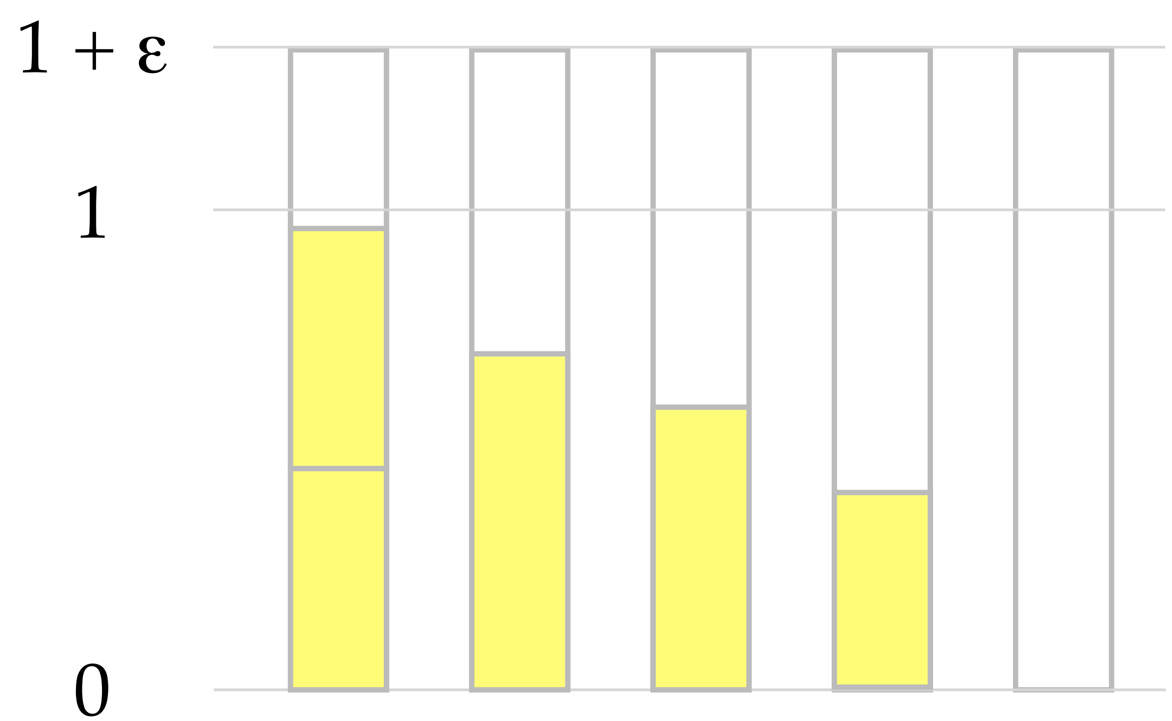
Superoptimale Aufteilung



ε-Dualer Approximations-Algorithmus für Bin-Packing

- Eingabe: n Objekte
- Ausgabe: Anzahl benötigter Behälter

Suboptimale Aufteilung



ε-Dualer Approximations-Algorithmus für Bin-Packing

Eingabe: $s_1, \ldots, s_n \in (0,1]$; Ausgabe $b \in \mathbb{N}$

1. Superopt. Bin-Packing für große Objekte > ε

   Abrunden auf eine von $s := \lceil \frac{1}{\epsilon^2} \rceil$ Größen

   Optimal lösen

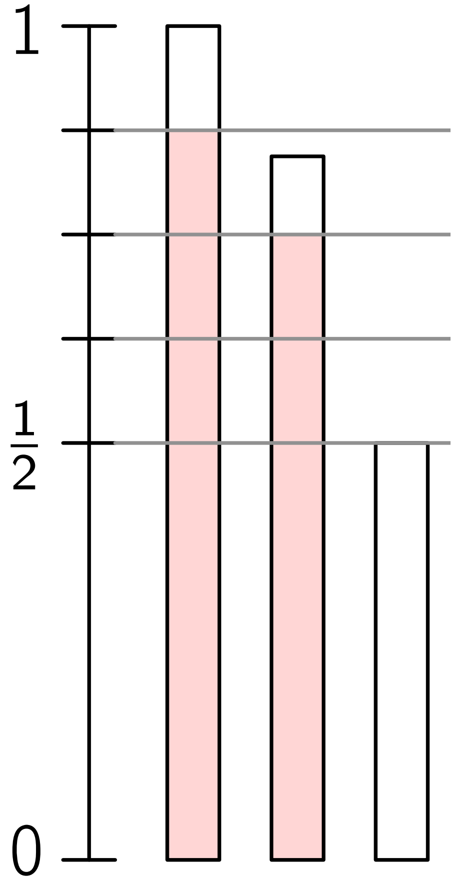
   Zurück zu Orginalgewichte (nun superopt.)

2. Kleine Gewichte einfügen (neue Behälter, falls nötig)

3. $b := \max\{b, \lceil \sum_{i \in [n]} s_i \rceil\}$

Beispiel für ε = ½:

s = 4 Größen

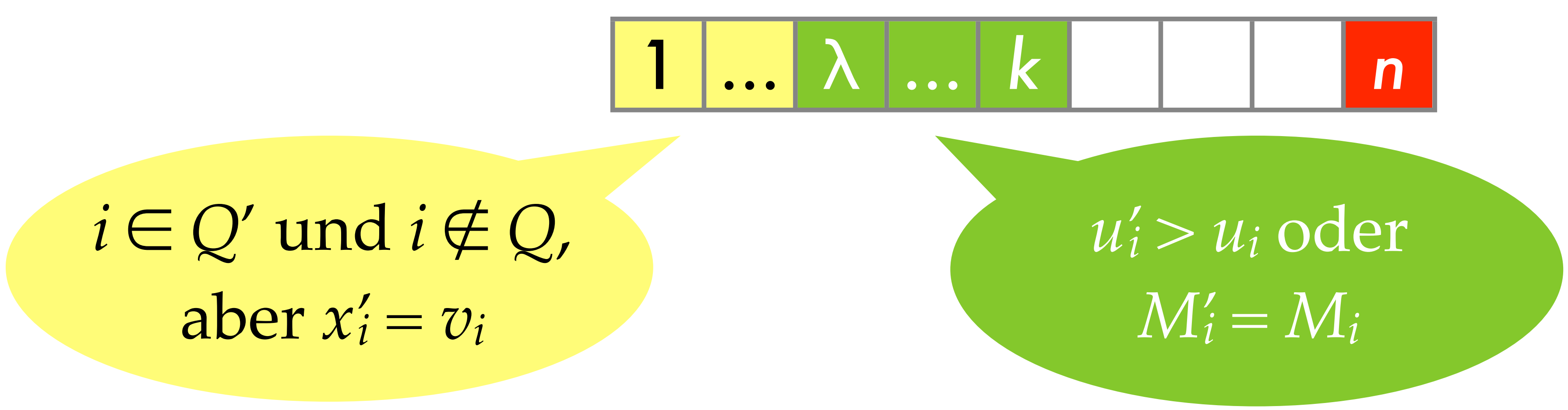


k-GSP $\neq$ GSP

- Beispiel mit 3 Spielern
  - Schwellwerte $\theta_1 = \theta_2 = 1 \text{ and } \theta_3 = \begin{cases} 2 & \text{if } b_1 > 1 \text{ and } b_2 > 1 \\ 1 & \text{else} \end{cases}$
  - Wenn $\boldsymbol{b} = (1, 1, 1)$, nehme alle auf
  - Andernfalls nehme indifferente Spieler nicht auf

2-GSP + Separabel ⇔ GSP

- Lemma: 2-GSP + separabel ⇒ Pareto-optimal
- Annahme: Mechanismus ist $(k-1)$-GSP, aber es gibt erfolgreiche Koalition K der Größe k
- Durch Lemma: $k < n$ und OE: $u_n' < u_n$
- Spieler in K wechseln nacheinander zu neuen Geboten



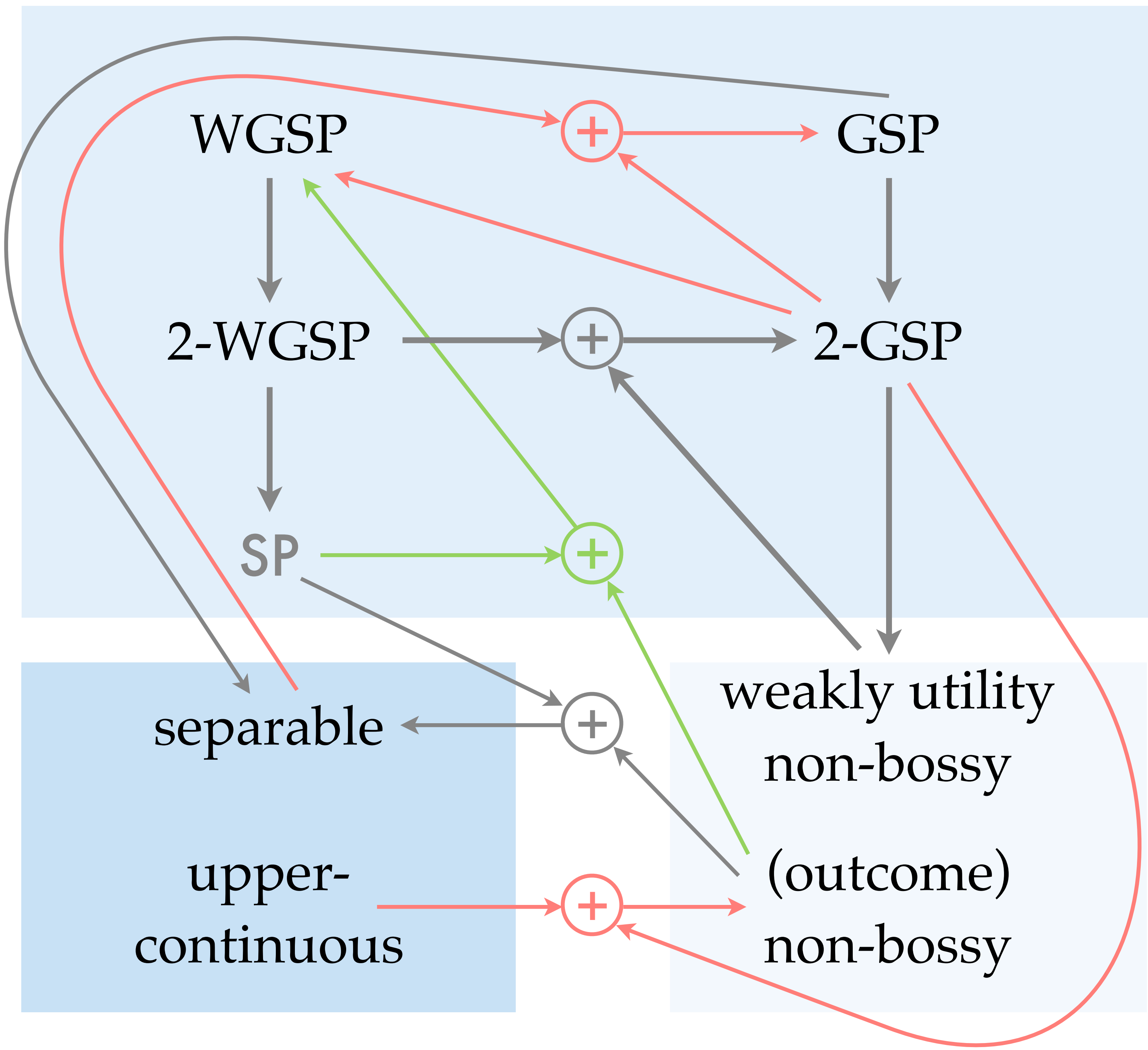
2-GSP + Separabel ⇒ Pareto-optimal

- Annahme: Allokation Q' ist besser als Q
  - OE: $Q' = \{\, m+1, \ldots, n \,\}$

1		*m*	…		*n*

- Def. $\boldsymbol{b}$ durch $b_i = b^\infty$ falls $i \in Q'$ und $b_i = -1$ sonst
- Spieler wechseln der Reihe nach zu $\boldsymbol{b}$
  - Definiere $\boldsymbol{b}^{\,i} := (b_1, \ldots, b_i, v_{i+1}, \ldots, v_n)$
  - Sei p be letzter Spieler mit $u_p(\boldsymbol{b}^{\,p}) < u_p(\boldsymbol{b}^{\,p-1})$
  - Dann: p bekommt Service, verliert Nutzen: $u_p(\boldsymbol{b}^{\,p}) < 0$
  - Dann auch $u_p(\boldsymbol{b}) < 0$, Widerspruch zu Pareto-Optimalität

Übersicht der Implikationen



Kapitel 5

Implikation per Definition

Mutuswami, Economics Letters ’05

Übersicht Optimierungs-Probleme

	Moulin		Azyklisch/Egalitär	
	BB	EFF	BB	EFF
$Q\|\|C_{max}$	$2\,d$	$2\,d \cdot (H_n + 1)$	2	$2\,H_n$
Vertex Cover	$\Omega(n^{1/3})$	$\Omega(n^{1/3})$	2	$\Theta(\log n)$
Facility Location	3	$\Theta(\log n)$	1,61	$\Theta(\log n)$
Steiner Tree	2	$\Theta(\log^2 n)$	2	$O(\log^2 n)$

- Moulin-Werte sind insb. auch untere Schranken!
- Alle obigen Probleme sind subadditiv
  - Untere Schranken für azyklische/egalitäre: 1-BB & H_n-EFF

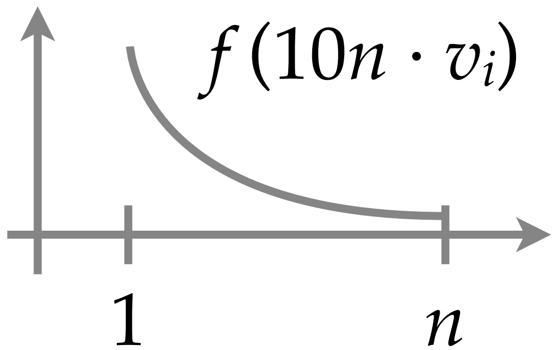
Untere Schranke: Effizienz

(Dobzinski et al., SAGT'08)

- Probabilistische Methode: ∃ schlechte Instanz

$$\Pr\left[v_i \le \tfrac{t}{10n}\right] = \tfrac{1}{n} + \int_1^t \tfrac{1}{z^2}\, dz$$

$$\mathrm{E}\left[10n \cdot v_i\right] = \ln n \qquad \mathrm{E}\left[\sum v_i\right] = \tfrac{\ln n}{10}$$



- Tschebyscheff: $\Pr\left[\frac{\ln n}{10} - 1 \le \sum v_i\right] \ge 0.99$
- Erwartete Zahlung von Spieler i: $\theta \cdot \Pr[v_i > \theta] \le \frac{1}{10n}$
- Wegen Budget-Balancierung: $\Pr[Q = \emptyset] \ge 0.9$
- Union Bound:

$$\Pr\left[\frac{\ln n}{10} - 1 \le \sum v_i \text{ und } Q = \emptyset\right] \ge 1 - 0.01 - 0.1 = 0.89$$

Unterschiedliche Teilnahme-Stufen

- Spieler können bis zu L Service-Level erhalten
- Statt Q nun Multimenge gesucht, also $\boldsymbol{q} \in [L]^n$
  - Gebote für jedes Level, also $\boldsymbol{b}_i \in \mathbf{R}^L$ für alle Spieler i
  - Fallende marginale Wertschätzungen $b_{i,1} \geq b_{i,2} \geq \ldots$
  - Marginale Kostenaufteilungs-Methode $\chi_{i,l}(\boldsymbol{q})$

$\boldsymbol{q} := (L, \ldots, L)$

solange es einen Spieler i gibt mit $b_{i,l} < \chi_{i,l}(\boldsymbol{q})$ für $0 < l \leq q_i$

$\quad q_i := q_i - 1$

Marginale Kostenaufteilungs-Methoden

- Mechanismus ist GSP, wenn:
  - cross-monoton: $\chi_{i,l}(\boldsymbol{s}) \leq \chi_{i,l}(\boldsymbol{t})$ für $\boldsymbol{s} \leq \boldsymbol{t}$ und $l < s_i$
  - nicht-fallend: $\chi_{i,1}(\boldsymbol{s}) \leq \ldots \leq \chi_{i,l}(\boldsymbol{s})$ für $l < s_i$
  - Level-beschränkt: $\chi_{i,l}(\boldsymbol{s}) = \chi_{i,l}(\boldsymbol{s}^{\leq l})$ wobei $\boldsymbol{s}^{\leq l} = (\min\{s_i, l\})_{i=1,\ldots,n}$

$\boldsymbol{q} := (L, \ldots, L)$

solange es einen Spieler i gibt mit $b_{i,l} < \chi_{i,l}(\boldsymbol{q})$ für $0 < l \leq q_i$

$\quad q_i := q_i - 1$