

Cost Sharing

Christian Hofmann

Wolfgang Mulzer, Yannik Stein

1 Kooperative Spiele und Kostenverteilung

Kostenverteilung beschäftigt sich grundlegend damit, wie man Zusammenarbeit von Spielern durchsetzen und aufrechterhalten kann.

Man spricht dabei von einer Menge $\mathcal{A}$ von Agenten, die kooperieren wollen, um einen Service zu erhalten und dabei ihren Gewinn zu maximieren. Grundsätzlich geht man davon aus, dass der Wert des erhaltenen Services für alle identisch ist, was bedeutet, dass es sich um ein Problem zur Minimierung der Kosten zum Erhalt handelt. Zur Verteilung der Kosten auf die Agenten definiert man eine Funktion $c : \mathcal{A} \times \mathcal{P}(\mathcal{A}) \rightarrow \mathbb{R}$. Ein Spiel im Sinne der Kostenverteilung wird repräsentiert durch ein Tupel $(\mathcal{A}, c)$.

1.1 Das Facility Location Game

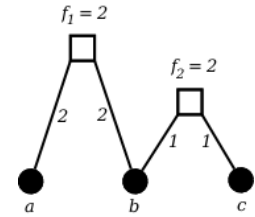
Als einfaches Beispiel für die Problematik wird das Facility Location Game verwendet. Hierbei handelt es sich um eine Menge von Agenten $\mathcal{A}$, $\mathcal{F}$ als Menge von Einrichtungen mit f_i als Eröffnungskosten für alle $i \in \mathcal{F}$, sowie Distanzkosten d_{ij} für jedes Tupel $(i, j) \in (\mathcal{A} \cup \mathcal{F})^2$. Es wird angenommen, dass die Distanz ein metrischer Raum ist.

Die Kosten für eine Koalition $S \subseteq \mathcal{A}$ ist folgendermaßen definiert:

$$c(S) = \min_{F' \subseteq \mathcal{F}} \left\{ \sum_{i \in F'} f_i + \sum_{j \in S} \min_{i \in F'} d_{ij} \right\}$$

Anhand des Beispiels stellen sich die Kosten folgendermaßen zusammen:

$$\begin{aligned} c(\{a\}) &= 4, & c(\{b\}) &= 3, & c(\{c\}) &= 3, \\ c(\{a, b\}) &= 6, & c(\{b, c\}) &= 4, & c(\{a, c\}) &= 7, & c(\{a, b, c\}) &= 8 \end{aligned}$$



2 Kern einer Kostenverteilung

Die Kosten einer Koalition können beliebig auf die teilnehmenden Agenten aufgeteilt werden. Es liegt nahe, dass ein Agent sich nur derjenigen Koalition anschließt, für die er die geringsten Kosten hat. Dies führt zur Definition eines Kerns, in der all jene Verteilungen liegen, bei denen sich kein Agent verbessern kann.

Definition 1. Sei $(\mathcal{A}, c)$ ein Spiel im Sinne der Kostenverteilung, dann ist der Vektor $\alpha \in \mathbb{R}^{\mathcal{A}}$ im Kern dieses Spiels, wenn es folgende Eigenschaften erfüllt:

- **Budget-Gleichgewicht:** $\sum_{j \in \mathcal{A}} \alpha_j = c(\mathcal{A})$
- **Kern-Eigenschaft:** $\forall S \subseteq \mathcal{A} : \sum_{j \in S} \alpha_j \leq c(S)$

Gäbe es nun in dem Beispiel des Facility Location Games eine weitere Einrichtung mit $f_3 = 3$ und Distanz 1 zu den Agenten a und c , würde sich $c(\{a, c\})$ auf 5 verringern. Für einen Vektor α des Kerns dieses neuen Spieles müsste gelten:

$$\alpha_a + \alpha_b \leq c(\{a, b\}) = 6, \quad \alpha_b + \alpha_c \leq c(\{b, c\}) = 4, \quad \alpha_a + \alpha_c \leq c(\{a, c\}) = 5$$

Durch Addition und anschließender Halbierung der 3 Ungleichungen erhält man $\alpha_a + \alpha_b + \alpha_c \leq 7,5 < c(\{a, b, c\})$. α kann das Budget-Gleichgewicht nicht erfüllen, daher ist der Kern leer.

Der Satz von Bondareva-Shapley liefert eine notwendige und hinreichende Bedingung für einen nicht-leeren Spiel-Kern. Hierfür wird folgende Definition benötigt:

Definition 2. Ein Vektor λ , der allen $S \subseteq \mathcal{A}$ nicht-negative Werte λ_S zuteilt heißt *balanced collection of weights*, wenn für alle $j \in \mathcal{A}$ gilt $\sum_{S:j \in S} \lambda_S = 1$.

Satz 3. Ein Spiel $(\mathcal{A}, c)$ im Sinne der Kostenverteilung hat einen nicht-leeren Kern genau dann, wenn für jede *balanced collection of weights* λ gilt $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq c(\mathcal{A})$.

Beweis. Nach Definition des Kerns hat das Spiel $(\mathcal{A}, c)$ genau dann einen nicht-leeren Kern, wenn die Lösung folgenden linearen Programms genau $c(\mathcal{A})$ ist.

$$\begin{array}{ll} \text{Maximiere} & \sum_{j \in \mathcal{A}} \alpha_j \\ \text{mit} & \forall S \subseteq \mathcal{A} : \sum_{j \in S} \alpha_j \leq c(S) \end{array}$$

Das duale Programm dafür ist

$$\begin{array}{ll} \text{Minimiere} & \sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \\ \text{mit} & \forall j \in \mathcal{A} : \sum_{S:j \in S} \lambda_S = 1 \\ & \forall S \subseteq \mathcal{A} : \lambda_S \geq 0 \end{array}$$

Die Nebenbedingungen des λ im dualen Programm entsprechen der aus der oben stehenden Definition. Zuletzt stellt das primale Programm in der Nebenbedingung sicher, dass $c(\mathcal{A})$ nie unterschritten wird. $\square$

Ein einfaches Beispiel ist das Facility Location Game mit der dritten Einrichtung. Ein möglicher Vektor λ ist folgender: $\lambda_{\{a,b\}} = \lambda_{\{b,c\}} = \lambda_{\{a,c\}} = 0,5$ und $\lambda_S = 0$ für alle anderen S . Hier entspricht $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S)$ exakt dem, was oben beschrieben wurde.

2.1 Approximierter Kern

Da es sehr häufig vorkommen kann, dass der Kern eines Spiels leer ist, und die Kostenfunktion schwierig zu berechnen ist, ist es oft erstrebenswert, unter Umgehung der Kostenfunktion festzustellen, ob ein Kern leer ist.

Definition 4. Ein Vektor $\alpha \in \mathbb{R}^{\mathcal{A}}$ ist im γ -approximiertem Kern (oder γ -Kern), wenn er neben der Kern-Eigenschaft folgendes erfüllt: **γ -Budget-Gleichgewicht:** $\gamma c(\mathcal{A}) \leq \sum_{j \in \mathcal{A}} \alpha_j \leq c(\mathcal{A})$

Der Satz von Bondareva-Shapley kann leicht für γ -Kerne generalisiert werden.

Satz 5. Ein Spiel $(\mathcal{A}, c)$ hat einen nicht-leeren γ -Kern genau dann, wenn für jede *balanced collection of weights* λ gilt $\sum_{S \subseteq \mathcal{A}} \lambda_S c(S) \geq \gamma c(\mathcal{A})$.

3 Gruppenstrategiesichere Mechanismen und kreuz-monotone Kostenverteilungsschemata

Für diesen Abschnitt betrachten wir als Problem einen Service-Anbieter mit einer Menge von Kunden. Wenn man davon ausgeht, dass die Nachfrage vom Preis abhängig ist, besteht für den Anbieter die Möglichkeit, potentielle Kunden in Abhängigkeit von Kosten und Zahlungsbereitschaft mittels eines Auktionsverfahrens auszuwählen. Das Ziel ist, einen Auktionsmechanismus zu entwerfen, der ein wahrheitsgemäßes bieten der Agenten fördert.

Sei $\mathcal{A}$ eine Menge von n Agenten, die den Service beziehen möchten und $c : \mathcal{P}(\mathcal{A}) \rightarrow \mathbb{R} \cup \{0\}$ die Kosten, die benötigt werden, um eine Menge $S \subseteq \mathcal{A}$ mit dem Service zu versorgen. Jeder Agent i hat einen Wert $u_i \in \mathbb{R}$, falls er den Service erhält. Dies ist damit gleichzeitig das Maximum dessen, was Agent i bereit ist zu zahlen.

Ein *Kostenverteilungsmechanismus* ist ein Algorithmus, der auf Basis von Geboten b_i aller Agenten $i \in \mathcal{A}$ entscheidet, welche Agenten den Service erhalten ($Q(b) \subseteq \mathcal{A}$) und wie viel sie jeweils zu bezahlen haben ($p(b) \in \mathbb{R}^n$). Bei Eindeutigkeit werden $Q(b)$ und $p(b)$ durch Q und p abgekürzt und weiterhin sei $q_i = 1$ wenn $i \in Q$ und 0 sonst. Der Mechanismus soll neben γ -Budget-Gleichgewicht folgende Bedingungen erfüllen:

- kein positiver Transfer: $\forall i \in Q : p_i \geq 0$
- freiwillige Teilnahme: $\forall i \in \mathcal{A} : (i \notin Q \implies p_i = 0) \wedge p_i \leq b_i$
- Konsumentensouveränität: $\forall i \in \mathcal{A} \exists b_i^* : p_i = b_i^* \implies i \in Q$

Wir suchen nach Mechanismen, genannt gruppenstrategiesicher, wenn er darüber hinaus noch folgende Eigenschaft hat. Sei $S \subseteq \mathcal{A}$ eine Koalition von Agenten und u, u' 2 Vektoren von Geboten mit $u_i = u'_i$ für alle $i \in S$ und (Q, p) und (Q', p') die Resultate des Mechanismus, wenn u bzw u' geboten wurden. Der Mechanismus soll nun für jede Koalition S für alle Agenten $i \in S$ für die die Ungleichung $u_i q'_i - p'_i \geq u_i q_i - p_i$ gilt auch die entsprechende Gleichung erfüllen.

Moulin hat gezeigt, dass die Eigenschaft der Kreuz-Monotonie als Kriterium verwendet werden kann, um einen gruppenstrategiesicheren Mechanismus zu entwickeln. Diese Eigenschaft besagt, dass Agenten nicht dadurch benachteiligt werden dürfen, dass Q wächst. Zur Definition dieser Eigenschaft benötigt man zunächst die Definition eines Kostenverteilungsschemas.

Definition 6. Sei $(\mathcal{A}, c)$ ein Spiel, dann ist ein Kostenverteilungsschema eine Funktion, die eine Kostenverteilung für jedes $S \subseteq \mathcal{A}$ liefert.

Formal: $\xi : \mathcal{A} \times \mathcal{P}(\mathcal{A}) \rightarrow \mathbb{R}$ für das gilt $\forall S \subseteq \mathcal{A} \forall i \notin S : \xi(i, S) = 0$.

Definition 7. Ein Kostenverteilungsschema ξ ist kreuz-monoton, wenn für alle $S, T \subseteq \mathcal{A}$ und alle $i \in S$ gilt $\xi(i, S) \geq \xi(i, S \cup T)$.

	Moulins gruppenstrategiesicherer Mechanismus
Mechanism $\mathcal{M}_\xi$	
Initilize $S \leftarrow \mathcal{A}$	
Repeat	
Let $S \leftarrow \{i \in S : b_i \geq \xi(i, S)\}$	
Until for all $i \in S, b_i \geq \xi(i, S)$	
Return $Q = S$ and $p_i = \xi(i, S)$ for all i	

Satz 8. Für jedes ξ als kreuz-monotonem Kostenverteilungsschema mit γ -Budget-Gleichgewicht ist $\xi(\cdot, \mathcal{A})$ im γ -Kern des Spiels.

Beweis. Man muss nur zeigen, dass $\xi(\cdot, \mathcal{A})$ die Kern-Eigenschaft erfüllt, also für jedes $S \subseteq \mathcal{A}$: $\sum_{i \in S} \xi(i, \mathcal{A}) \leq c(S)$. Durch die Kreuz-Monotonie gilt $\forall i \in S : \xi(i, \mathcal{A}) \leq \xi(i, S)$. Daraus und durch das γ -Budget-Gleichgewicht folgt $\sum_{i \in S} \xi(i, \mathcal{A}) \leq \sum_{i \in S} \xi(i, S) \leq c(S)$. $\square$

Satz 9. Für jedes ξ als kreuz-monotonem Kostenverteilungsschema mit γ -Budget-Gleichgewicht ist $\mathcal{M}_\xi$ gruppenstrategiesicher.

Beweis. Angenommen, es gäbe eine Koalition T von Agenten, die davon profitieren, u' anstelle ihres wahren Wertes u zu bieten, dann kann man T in 2 Gruppen aufteilen: T^+ und T^- , je nachdem, ob ihr Gebot in u' größer ist als in u oder nicht. Zunächst können wir sagen, dass T^+ leer ist, da jeder Agent im Falle eines Verlustes nicht profitiert und im Falle eines Gewinnes mehr zahlen muss, als ihm der Service Wert ist.

Seien S' und S nun die Gewinner bezüglich der Gebote in u' und u . Da jedes Gebot in u' höchstens so hoch sein kann wie in u , könnten einige Agenten aus T nicht gewinnen, daher gilt $S' \subseteq S$. Die Kreuz-Monotonie impliziert nun aber, dass $\forall i \in S' : \xi(i, S') \geq \xi(i, S)$, wodurch sich keine Verbesserung ergeben kann. $\square$

4 Submodulare Spiele

Definition 10. Ein Spiel wird submodular genannt, wenn die Kostenfunktion c folgende Eigenschaft erfüllt: $\forall S, T \subseteq \mathcal{A} : c(S) + c(T) \geq c(S \cup T) + c(S \cap T)$.

Diese Definition ist äquivalent zu der Bedingung, dass für alle Agenten i, j und jede Koalition $S \subset \mathcal{A} \setminus \{i, j\}$ die Kosten zur Aufnahme von i zu S nicht kleiner ist als zur Aufnahme von i zu $S \cup \{j\}$.

Eine submodulare Kostenfunktion sorgt dafür, dass die Lösung des primalen linearen Programmes immer $c(\mathcal{A})$ ist. Sei α eine mögliche Lösung für dieses Programm, dann nennen wir eine Menge $S \subseteq \mathcal{A}$ fest, wenn die entsprechende Ungleichung im Programm fest ist, das heißt $\sum_{j \in S} \alpha_j = c(S)$.

Lemma 11. Sei α eine mögliche Lösung für das lineare Programm. Wenn 2 Mengen $S_1, S_2 \subseteq \mathcal{A}$ fest sind, dann ist es auch $S_1 \cup S_2$.

Beweis. Da α eine mögliche Lösung ist, gilt $\sum_{j \in S_1 \cap S_2} \alpha_j \leq c(S_1 \cap S_2)$. Dies impliziert zusammen mit der Submodularität von c und der Tatsache, dass S_1 und S_2 fest sind folgendes:

$$\begin{aligned} c(S_1 \cup S_2) &\leq c(S_1) + c(S_2) - c(S_1 \cap S_2) \\ &\leq \sum_{j \in S_1} \alpha_j + \sum_{j \in S_2} \alpha_j - \sum_{j \in S_1 \cap S_2} \alpha_j \\ &= \sum_{j \in S_1 \cup S_2} \alpha_j \end{aligned}$$

Daraus folgt: $S_1 \cup S_2$ ist fest. $\square$

Folgerung 12. Es gibt eine eindeutige maximale feste Menge, die aus der Vereinigung aller festen Mengen besteht.

Algorithmus zur Bestimmung einer Kostenverteilung

Algorithm SubmodularCostShare

Input: submodular cost sharing game $(\mathcal{A}, c)$
 set $T \subseteq \mathcal{A}$ of agents that receive the service

Output: cost shares α_j for every $j \in T$

For every j , initialize α_j to 0
 Let $F = \emptyset$
 While $T \setminus F \neq \emptyset$ do
 Increase all α_j for $j \in T \setminus F$ at the same rate
 until a new set goes tight
 Let F be the maximal tight set

Satz 13. *Das Kostenverteilungsschema durch oben gegebenen Algorithmus ist kreuz-monoton.*

Beweis. Seien $T_1 \subset T_2 \subseteq \mathcal{A}$, dann führen wir den Algorithmus gleichzeitig für T_1 und T_2 und zeigen, dass zu jedem Zeitpunkt die Menge der fixierten Elemente aus dem T_1 -Durchlauf eine Teilmenge derer des T_2 -Durchlaufs ist.

Man betrachtet den Zeitpunkt t , bei dem α^1 bzw α^2 die Werte der Variablen und F_1 bzw F_2 die Mengen der fixierten Elemente sind.

$$\begin{aligned}
 c(F_1 \cup F_2) &\leq c(F_1) + c(F_2) - c(F_1 \cap F_2) \\
 &\leq \sum_{i \in F_1} \alpha_i^1 + \sum_{i \in F_2} \alpha_i^2 - \sum_{i \in F_1 \cap F_2} \alpha_i^1 \\
 &= \sum_{i \in F_1 \setminus F_2} \alpha_i^1 + \sum_{i \in F_2} \alpha_i^2 \\
 &\leq \sum_{i \in F_1 \cup F_2} \alpha_i^2
 \end{aligned}$$

Die erste Ungleichung folgt von der Submodularität von c , die zweite dadurch, dass die F fest sind in Bezug auf α und die letzte durch die Tatsache, dass für alle $i \in F_1 \setminus F_2$, da $i \in T_1 \subset T_2$ in T_2 zum Zeitpunkt t nicht fixiert ist, gilt $\alpha_i^2 = t \geq \alpha_i^1$. Dies impliziert, dass $F_1 \cup F_2$ fest ist in Bezug auf α^2 . Da F_2 nach Definition die maximale feste Menge in Bezug auf α^2 ist, muss gelten $F_1 \cup F_2 = F_2$, womit gilt $F_1 \subseteq F_2$. □

Quelle: Nisan, Roughgarden, Tardos, Vazirani: Algorithmic Game Theory, Kapitel 15.