Bidding and Allocation in Combinatorial Auctions

1. Introduction

1.1. Motivation

Gestiegene Popularität von Auktionen: Internet, Privatisierungen, Agentensysteme, E-Commerce

Problem: Auktionen nicht-identischer Gegenstände. Wert eines Gegenstandes für einen Bieter jeweils abhängig vom Erhalt anderer ergänzender Gegenstände

Wunsch nach kombinatorischen Auktionen, jedoch Problem wegen exponentiell vieler Kombinationen

1.2. Grundlegendes

Gebote: Verpflichtet zum maximalen Betrag, den ein Bieter bereit ist für jede mögliche Kombination von Gegenständen. Protokolle

Zuteilung: Dabei Optimierung eines Zielfunktion, bspw. Gewinn des Auktionators oder gesamtwirtschaftliche Effizienz

Zahlung: Wie viel muss jeder Bieter bezahlen?

Strategie: Gebotsstrategie. Sinnvoll: Gewünschtes Auskommen, selbst wenn jeder eigensinnig handelt

Konzentration in dieser Arbeit auf Gebote (im „verschlossenen Umschlag") und Zuteilung. Abwägung: Ausdrucksfähigkeit (wobei „wichtige" Arten von Gebote einfach auszudrücken sein sollen) $\leftrightarrow$ Einfachheit

1.3. Vorhergehende Arbeiten

Kombinatorische Auktionen selten benutzt. Abhilfen: Ignorieren, Wiederverkaufsplattformen, Englische (steigende) Auktionen mit mehreren Runden, Bieter dürfen Teams bilden

Wenn Zahlungen nach Vickrey-Clarke-Groves Mechanismus und Zuteilungsalgorithmus perfekt, ist „Ehrlichkeit" dominante Strategie für Bieter.

1.4. Ergebnisse

Gebotssprachen: OR, XOR, OR-of-XORs, XOR-of-ORs, OR*

Zuteilung mit linearer Programmierung. Wenn Gebote bestimmte Bedingungen erfüllen, ergeben sich ganzzahlige Werte, also eine optimale Zuteilung

2. Modell

m zu versteigernde Objekte, n Bieter, jeder Bieter i hat private Bewertungsfunktion $v_i : \mathcal{P}([m]) \to \mathbb{R}_{\geq 0}$ (keine weiteren Abhängigkeiten!). Weitere Eigenschaften von v_i: Kostenloses Weglassen: $v_i(S) \leq v_i(T)$ für $S \subseteq T$, Normalisierung: $v_i(\emptyset) = 0$.

Definition: Für Bieter i, zwei disjunkte Mengen S und T sind komplementär, wenn $v_i(S \mathbin{\dot\cup} T) > v_i(S) + v_i(T)$, und Substitute, falls $v_i(S \mathbin{\dot\cup} T) < v_i(S) + v_i(T)$.

Auktionator sucht paarweise disjunkte $S_1, \ldots, S_n$ so dass $\sum_{i=1}^n v_i(S_i)$ maximiert wird.

3. Gebotssprachen

3.1. Beispiele

Eine symmetrischen Bewertungsfunktion schätzt alle Artikel vom Wert identisch ein.

Additiv: $v(S) = |S|$

Single Item: $v(S) = 1$ falls $S \neq \emptyset$

k-Budget: $v(S) = \min\{k, |S|\}$

Majority: $v(S) = 1$ falls $|S| \geq \frac{m}{2}$, 0 sonst

Allgemein symmetrisch: p_j bezeichnet Wert für j-tes gewonnenes Objekt. $v_i(S) = \sum_{j=1}^{|S|} p_j$.

Fallend symmetrisch: Wie zuvor, $p_j \geq \cdots \geq p_1$

Assymmetrische Bewertungsfunktionen:

Monochromatisch: S setzt sich aus Gegenständen zwei verschiedenen Arten zusammen, d. h. $|S| = l + k$. Dann $v_i(S) = \max\{l, k\}$.

Eins-von-jeder-Sorte: Insg. $\frac{m}{2}$ Paare. Bieter möchte 1 von jedem Paar. Wenn S nun k Paare und l einzelne Objekte enthält ($|S| = 2k + l$), so ist $v_i(S) = k + l$.

Grundlegende Gebotssprachen

Elementargebot (atomic bid) (S, p), wobei $S \subseteq [m]$ und p der gebotene Preis ist.

OR-Gebote Menge von Elementargeboten. Ist (sofern Mengen disjunkt) bereit, alles zu nehmen.

XOR-Gebote Menge von Elementargeboten. Bieter will nu eine Menge davon.

Satz: OR-Gebote können genau alle Bewertungsfunktionen ohne Substitute darstellen. XOR-Gebote können beliebige Bewertungsfunktionen darstellen.

Definition: Die Größe eines Gebots ist die Anzahl enthaltener Elementargebote

Satz: Eine additive Bewertungsfunktion hat als XOR-Gebot Größe 2^m.

3.3. OR-of-XORs-Gebote

Satz: OR-of-XORs-Gebote können fallend symmetrische Bewertungsfunktionen in Größe m^2 ausdrücken.

Satz: Monochromatische Bewertungsfunktionen benötigt mindestens Größe $2 \cdot 2^{m/2}$ bei OR-of-XORs-Geboten.

3.4. XOR-of-ORs-Gebote

Satz: $k := \sqrt{m}/2$. Dann benötigt eine k-Budget Bewertungsfunktion als XOR-of-ORs-Gebot mindestens Größe $2^{m^{1/4}}$.

3.5. OR/XOR-Formeln

Definition: Seien v, u Bewertungsfunktionen. Dann sind $(v \,\mathrm{XOR}\, u)(S) := \max\{v(S), u(s)\}$ und $(v \,\mathrm{OR}\, u)(S) = \max_{R,T \subseteq S, R \cap T = \emptyset}\{v(R) + u(T)\}$ Bewertungsfunktionen.

3.6. Gebotssprachen und polynomielle Berechenbarkeit

Definition: Eine Gebotssprache heißt polynomiell interpretierbar, wenn es einen Polynomialzeit-Algorithmus gibt, der zu jedem Gebot b dieser Sprache und jedem $S \subseteq [m]$ den Wert $b(S)$ berechnet.

Lemma: Nur Elementargebote und XOR-Gebote sind polynomiell interpretierbar. Insb. OR ist es nicht!

Definition: Eine Funktion V mit Eingaben b, S und Zertifikat w verifiziert eine Gebotssprache, falls $\max_w\{V(b, S, w)\} = b(S)$. Eine Gebotssprache heißt polynomiell verifizierbar, wenn es einen polynomiellen Algorithmus für eine solche Funktion V gibt.

Lemma: Alle bisherigen Gebotssprachen (und auch OR*) sind polynomiell verifizierbar.

Applet Gebote Ein Gebot b ist ein Polynomialzeit-Programm b_{app}, das eine Teilmenge S und einen String w übergeben bekommt. Es gilt $b(S) = \max_w\{b_{app}(S, w)\}$.

4. OR Gebote mit Phantomobjekten

OR*-Gebote Menge von Elementargeboten, wobei zusätzlich jeder Bieter eigene Phantomobjekte verwenden darf (um dadurch exklusives Oder zu simulieren).

Lemma: Jedes OR-of-XORs- (bzw. XOR-of-ORs-) Gebot der Größe s kann auch als OR*-Gebot der Größe s dargestellt werden, unter Benutzung von maximal s (bzw. s^2)Phantomobjekten.

Satz: Jede OR/XOR-Formeln der Größe s kann auch als OR*-Gebot der Größe s dargestellt werden, unter Benutzung von maximal s^2 Phantomobjekten.

Lemma: Die Majority-Bewertungsfunktion benötigt für die Darstellung als OR*-Gebot mindestens Größe $\binom{m}{m/2}$.

Beachte: OR* könnte so erweitert werden, dass äußere Abhängigkeiten ausgedrückt werden: Mehrere Bieter könnten sich Phantomobjekte teilen.

5. Das Zuteilungsproblem und lineare Programmierung

5.1. Formalisierung des Zuteilungsproblems

Seien B_i, $i \in [n]$, die Elementargebote (S_i, p_i). (Es ist hier bei OR* egal, zu welchem Bieter sie gehören.) Dabei $S_i \subseteq [m]$ und $p_i \geq 0$. Es sollen die Zuschläge so verteilt werden, dass $\sum_{i \in Winners} p_i$ maximal wird und für alle $j \in [m]$ gilt: $|\{i \in Winners \mid j \in S_i\}| \leq 1$.

Als ganzzahliges Programm:

$$
\begin{array}{lll}
\text{max.} & \displaystyle\sum_{i=1}^{n} x_i p_i & \\
\text{gem.} & \displaystyle\sum_{i \in [n] \mid j \in S_i} x_i \leq 1 & \forall j \in [m] \\
 & x_i \in \{0, 1\} & \forall i \in [n]
\end{array}
$$

5.2. Relaxierung als lineares Programm

Ersetze letzte Bedingung druch $x_i \geq 0$. (Mehr ist aufgrund der anderen Bedingung und da $p_i = 0$ sein muss, falls $S_i = \emptyset$, nicht notwendig.)

5.3. Bedeutung der LP-Lösung

Fraktionale kombinatorische Auktion (Öl, Bandbreiten, etc.)

Definition: Eine Zuteilung $(x_1, \dots, x_n) \in \{0,1\}^n$ wird durch Einzelpreise y_i, $i \in [m]$, bestätigt, falls aus $x_i = 1$ (Zuschlag für B_i) folgt, dass $p_i \geq \sum_{j \in S_i} y_i$, und ferner $x_i = 0 \Rightarrow p_i \leq \sum_{j \in S_i} y_i$. Eine Zuteilung wird exakt bestätigt, wenn $x_i = 1 \Rightarrow p_i = \sum_{j \in S_i} y_i$.
Eine Zuteiltung heißt voll, wenn für alle $j \in [m]$ ein $x_i = 1$ mit $j \in S_i$ existiert. Eine Auktion lässt Einzelpreise zu, falls es eine volle Zuteilung gibt, die durch einen Vektor von Einzelpreisen bestätigt wird.

Satz: Eine kombinatorische Auktion lässt Einzelpreise genau dann zu, wenn die LP-Relaxierung nur ganzzahlige Lösungen hat. Die volle Zuteilung ist dann diese Lösung und die (exakt) bestätigenden Einzelpreise sind die Lösungen zum dualen linearen Programm.
Das duale lineare Programm:

$$\begin{aligned} \text{min.} \quad & \sum_{i=1}^{m} y_i \\ \text{gem.} \quad & \sum_{j \in S_i} y_i \geq p_i \quad && \forall i \in [n] \\ & y_j \geq 0 && \forall j \in [m] \end{aligned}$$

Lemma: Bieter 1: $(A, 5)\,\mathrm{XOR}(B, 6)$ und Bieter 2: $(B, 3)$ lässt ohne Phantomobjekte keine Einzelpreise zu.

6. Fälle, in denen die LP-Relaxierung optimal ist

6.1. Reihenfolge

Lemma: Können die Objekte so umbenannt werden, dass alle Gebote einen zusammenhängenden Bereich $\{k, k+1, \dots, l\} \subseteq [m]$ darstellen, ergibt die LP-Relaxierung eine optimale eine optimale Zuteilung.

Lemma: Hierarchische Gebote: Gilt für alle $S, T \subseteq [n]$, die zu Elementargeboten eines Bieters gehören, dass $S \cap T = \emptyset$, $S \subseteq T$ oder $T \subseteq S$, dann ist die LP-Relaxierung wiederum optimal.

6.2. Gegenseitiger Ausschluss

Satz: Weitere Fälle, in denen die LP-Relaxierung optimal ist: OR-of-XORs von Einzelgeboten (auf jeweils nur ein Objekt), Gebote für nicht mehr als ein Objekt (jedes Elementargebot ist von der Form $(\{j, g_i\}, p)$, wobei g_i ein eigenes Phantomobjekt für Bieter i ist). Ferner: Fallend symmetrische Gebote.

6.3. Teilstruktur

Definition: Eine Auktion heißt direkte Summe von Auktionen über die Partitionen Q, R, $S = Q \cup R$, falls alle Bewertungsfunktionen $v : S \to \mathbb{R}_{\geq 0}$ dargestellt werden könne als $v(S) = v\big|_Q(S \cap Q) + v\big|_R(S \cap R)$.

Lemma: Die direkte Summe zweier Auktionen, die ganzzahlige LP-Lösungen haben, weist ebenfalls diese Eigenschaft auf.

Definition: Ein Elementargebot (S_i, p_i) in einer Auktion wird durch die anderen Elementargebote majorisiert, falls $p_i < \sum_{j \in S_i} y_j$ und die y_j die Lösung des dualen LPs sind.

Lemma: Wenn eine Auktion ohne majorisierter Elementargebote eine ganzzahlige LP-Lösung hat, dann hat sie dies auch weiterhin nach Hinzufügen majorisierter Elementargebote.

7. Mit nicht-ganzahlige Lösungen umgehen

7.1. Ein greedy Algorithmus

Das primale LP weist nur den Elementargeboten (S_i, p_i) einen echt positiven Wert zu, bei denen $p_i = \sum_{j \in S_i} y_j$ der „faire" Preis ist (wie ihn das duale LP liefern würde).

Greedy-Zuteilungsalgorithmus:

LP-Relaxierung durchführen
Elementargebote (S_i, p_i) fallend nach $p_i / \sum_{j \in S_i} y_j$ ordnen. Gebote mit $x_i > 0$ (also $p_i = \sum_{j \in S_i} y_j$) werden fallend nach x_i sortiert.
$\mathit{Winners} := \emptyset$, $\mathit{AllocatedItems} := \emptyset$
for alle Elementargebote (S_i, p_i) in der neuen Reihenfolge **do**
    $\mathit{Winners} := \mathit{Winners} \cup \{i\}$
    $\mathit{AllocatedItems} := \mathit{AllocatedItems} \cup S_i$

Ein Branch-and-Bound Zuteilungsalgorithmus

Schranken für Optimalität der Zuteilung:
Obere: Die LP-Relaxierung
Untere: Der Greedy-Algorithmus

Branch-and-Bound Zuteilungsalgorithmus:

Berechne *upperBound* mit LP-Relaxierung
if *upperBound* $\leq$ *lowValue* **then**
    gebe Fehler zurück
Berechne *lowerBound* mit Greedy-Algorithmus
if *lowerBound* $>$ *lowValue* **then**

6: *lowValue* := *lowerBound*
7: Merke die eigentliche Lösung
8: (S, p) := erstes Elementargebot nach Reihenfolge wie im Greedy-Algorithmus
9: Probiere, dass (S, p) gewinnt:
   - Entferne (S, p) und alle Objekte aus S. Entferne alle Gebote, die Elemente aus S enthalten
   - Rufe Algorithmus rekursiv auf, wobei *lowValue* um p verringert
   - Wenn rekursiver Aufruf Erfolg hatte, aktualisiere *lowValue* durch Erhöhen des zurückgegebenen Wertes um p, merke die Lösung
10: Probiere, dass (S, p) verliert:
   - Entferne nur das Gebot (S, p)
   - Rufe Algorithmus rekursiv auf, mit aktuellem *lowValue*
   - Wenn rekursiver Aufruf Erfolg hatte, aktualisiere *lowValue* auf den zurückgegebenen Wert, merke die Lösung
11: Falls *lowValue* akutalisiert wurde, gebe Lösung mit Erfolg zurück, andernfalls gebe Fehler zurück

Lemma: Der Branch-and-Bound-Algorithmus gibt eine optimale Zuteilung zurück

Mögliche Optimierungen: Verwende nur fast-optimale obere Schranke, evtl. reicht ϵ-Optimalität, Merken von Lösungen von Teilproblem (allerdings Speicherbedarf), LP-Solver mit alter Lösung initialisieren

---

Selfish Routing in Capacitated Networks

1. Einführung

Wardrops erstes Prinzip (Nash-/Benutzer-Equilibrium, NE/BE):

> Die Reisezeit auf allen benutzten Routen ist gleich. Sie ist ferner geringer [*Anm.:* oder gleich] als die, die ein einzelnes Fahrzeug auf einem unbenutzten Pfad hätte.

Zweites Prinzip (Systemoptimum):

> Die durchschnittliche Reisezeit ist minimal.

Altes Ergebnis (Roughgarden, Tardos): Wenn nur stetige, nicht-fallende Latenzfunktionen, deren Produkt mit der Identität konvex ist, dann ist Preis der Anarchie unabhängig von der Netzwerk-Topologie.

2. Modellierung

2.1 Netzwerke ohne Kapazitätsbeschränkungen

Definition: Ein <u>Wardrop-Spiel ohne Kapazitätsbeschränkungen</u> besteht aus einem <u>Netzwerk</u> $(V, E, (\ell_e)_{e \in E})$ und n Routing-Anforderungen $(a_i, o_i, d_i)_{i \in n}$. Dabei: $\ell_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ sind stetige nicht-fallende <u>Latenzfunktionen</u>. Ferner: a_i ist ein Fluss (Bedarf), der von $o_i \in V$ nach $d_i \in V$ geroutet werden muss.

$\mathcal{P}_i := \{P \mid P \text{ ist Pfad von } o_i \text{ nach } d_i\}$. $\mathcal{P} := \bigcup_{i \in [n]} \mathcal{P}_i$.

Definition: Ein <u>Routing</u> (oder <u>Pfadfluss</u>) ist ein nicht-negativer Vektor $f := (f_P)_{P \in \mathcal{P}}$. Dabei heißt f <u>zulässig</u>, wenn es den Bedarf erfüllt: $\forall i \in [n] : \sum_{P \in \mathcal{P}_i} f_P = a_i$. Es bezeichne S die Menge aller zulässigen Routings.

Die Last auf einer Kante ist $l_e(f) := \sum_{P \in \mathcal{P} | P \ni e} f_P$. Abkürzend: $f_e := l_e(f)$. Reisezeit entlang eines Pfades ist $\ell_P(f) := \sum_{e \in P} \ell_e(l_e(f)) = \sum_{e \in P} \ell_e(f_e)$.

Definition: Die <u>(sozialen) Kosten</u> eines Routings sind definiert als Gesamtreisezeit: $C(f) := \sum_{P \in \mathcal{P}} \ell_P(f) f_P = \sum_{e \in E} \ell_e(f_e) f_e$.

Setze $\ell_e^f := \ell_e(f_e)$. Für $x \in S$ ist $C^f(x) := \sum_{P \in \mathcal{P}} \ell_P(f) x_P = \sum_{e \in E} \ell_e^f x_e$.

Nicht-lineares Optimierungsprogramm für das Systemoptimum (der Einfachheit halber mit exponentiell vielen Nebenbedingungen):

$$\begin{aligned} \text{max.} \quad & \sum_{e \in E} \ell_e(f_e) f_e & & \qquad (1) \\ \text{gem.} \quad & \sum_{P \in \mathcal{P} | P \ni e} f_P = f_e & \forall e \in E & \\ & \sum_{P \in \mathcal{P}_i} f_P = a_i & \forall i \in [n] & \\ & f_P \geq 0 & \forall P \in \mathcal{P} & \end{aligned}$$

2.2. Ergebnisse vorheriger Arbeiten

- $f \in S$ ist BE genau dann, wenn $\ell_P(f) \leq \ell_Q(f)$ für alle $i \in [n]$, $P, Q \in \mathcal{P}_i$ und $f_P > 0$.
- $f \in S$ ist BE genau dann, wenn $C(f) = C^f(f) \leq C^f(x)$ für alle $x \in S$.
- Ein BE existiert immer und kann effizient berechnet werden. Dazu ersetzt man (1) durch $\sum_{e \in E} \int_0^{f_e} \ell_e(x)\, dx$.
- $f \in S$ ist optimal genau dann, wenn $\sum_{e \in P} \ell_e^*(f_e) \leq \sum_{e \in Q} \ell_e^*(f_a)$ für alle $i \in [n]$, $P, Q \in \mathcal{P}_i$ und $f_P > 0$. Dabei $\ell_e^* := \ell_e + \ell_e' \cdot \text{id}$.

Satz: Sei f BE und f^* Optimum in einem Wardrop-Spiel mit affin-linearen Latenzfunktionen (und positiven Koeffizienten). Dann gilt $\frac{C(f)}{C(f^*)} \leq \frac{4}{3}$.

3. Netzwerke mit Kapazitätsbeschränkungen

Modellierung von Kapazitäten durch Latenzfunktionen, die nahe der Kapazitätsgrenze gegen unendlich streben, ist unrealistisch (empirisch gezeigt): Zu hohe Reisezeiten.

Definition: $(V, E, (f_e)_{e \in E}, (c_e)_{e \in E})$ ist ein Netzwerk mit Kapazitätsbeschränkungen, wenn $(V, E, (f_e)_{e \in E})$ ein Netzwerk ist und $c_e \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ der maximale Fluss auf Kante e ist.

Definition: Ein Pfad $P \in \mathcal{P}$ heißt saturiert, falls $f_e = c_e$ für ein $e \in \mathcal{P}$ gilt. Andernfalls unsaturiert.

Definition: $f \in S$ heißt BE mit Kapazitätsbeschränkungen (BEK), wenn $\ell_P(f) \leq \ell_P(f)$ für alle $i \in [n]$, $P, Q \in \mathcal{P}_i$, $f_P > 0$ und Q unsaturiert.

3.1. Eigenschaften von BEKs

Lemma: Im Allgemeinen haben BEKs unterschiedliche Qualität und ihr Menge ist nicht-konvex.

Lemma: Das Verhältnis zwischen einem BEK und dem Systemoptimum kann beliebig schlecht werden.

Bemerkungen: Ein BEK ließe sich einschränkender definieren um nicht plausible Equilibrien auszuschließen:

> Kein beliebig kleiner Teil von Reisenden auf gleichem Pfad kann seine Reisezeit durch Wechseln auf einen anderen Pfad verkleinern.

3.2. Das BMW-Equilibrium

Definition: Eine optimale Lösung zum folgenden nicht-linearen Optimierungsproblem heißt BMW-Equilibrium, kurz BMW.

$$
\begin{aligned}
\text{max.} \quad & \sum_{e \in E} \int_0^{f_e} \ell_e(x)\, dx \\
\text{gem.} \quad & \sum_{P \in \mathcal{P} | P \ni e} f_P = f_e \quad \forall e \in E \\
& \sum_{P \in \mathcal{P}_i} f_P = a_i \quad \forall i \in [n] \\
& f_e \leq c_e \quad \forall e \in E \\
& f_P \geq 0 \quad \forall P \in \mathcal{P}
\end{aligned}
$$

Lemma: $f \in S$ ist BMW genau dann, wenn $C(f) \leq C^f(x)$ für alle $x \in S$.

Satz: Jedes BMW ist ein BEK.

Modellierung mit Strafparameter: $\ell_e^\mu(x_e) := \begin{cases} \ell(x_a) + \frac{\mu}{c_e - x_e} & \text{für } x_e < c_e \\ \infty & \text{für } X_e \geq c_e \end{cases}$

Lemma: Im Grenzfall $\mu \to 0$ agieren Verkehrsteilnehmer bei einem BE genauso, wie sie es bei einem BEK würden. Anders ausgedrückt: Gegeben ein Wardrop-Spiel mit Kapazitätsbeschränkungen. Ist (μ_i) eine Nullfolge und (f^i) die entsprechende Folge von BE im selben Netzwerk, aber mit Latenzfunktionen ℓ_e^μ, so ist jeder Häufungspunkt von (f^i) ein BMW.

Es bezeichne C^μ die Gesamtkosten, falls die Latenzfunktionen ℓ_e jeweils durch ℓ_e^μ ersetzt werden.

Lemma: Sei $(\bar{f^i})$ eine Teilfolge von (f_i), die gegen ein BMW konvergiert (wie im vorigen Lemma beschrieben), $(\bar{\mu}_i)$ die entsprechende Teilfolge von (μ_i). Ein * bezeichne das jeweilige Systemoptimum. Es gilt im Allgemeinen:

$$\frac{C^{\bar{\mu}_i}(\bar{f}^i)}{C^{\bar{\mu}_i}(\bar{f}^{i,*})} \xrightarrow{\bar{\mu}_i \to 0} \!\!\!\!\!\!\!/\;\;\; \frac{C(f)}{C(f^*)}$$

3.3. Preis der Anarchie bei BMW-Equilibrien

Satz: Ist $f \in S$ BEK und x ein zulässiges Routing im gleichen Netzwerk, allerdings mit verdoppelten Kapazitäten und Anforderungen, so gilt: $C(f) \leq C(x)$.

Für eine Menge von Latenzfunktionen $\mathcal{L}$ und $\ell \in \mathcal{L}$ und $v \geq 0$ wird definiert:

$$
\begin{aligned}
\beta(v, \ell) &:= \begin{cases} 0 & \text{wenn } v = 0 \text{ oder } \ell(v) = 0 \\ \frac{1}{\ell(v) v} \cdot \max_{x \geq 0} \{x(\ell(v) - \ell(x))\} & \text{sonst} \end{cases} \\
\beta(\ell) &:= \sup_{v \geq 0} \beta(v, \ell) \\
\beta(\mathcal{L}) &:= \sup_{\ell \in \mathcal{L}} \beta(\ell)
\end{aligned}
$$

Lemma: Eigenschaften:

- $\beta(v, \ell) \geq 0$
- $\max_{x \geq 0}\{x(\ell(v) - \ell(x))\}$ wird auf $[0, v]$ angenommen
- $\beta(v, \ell) < 1$, $\beta(\ell) \leq 1$, $\beta(\mathcal{L}) \leq 1$

Satz: Sei $f \in S$ BEK, f^* Systemoptimum. Es gelte $\ell_e \in \mathcal{L}$ für alle $e \in E$. Dann $\frac{C(f)}{C(f^*)} \leq \frac{1}{1 - \beta(\mathcal{L})}$.

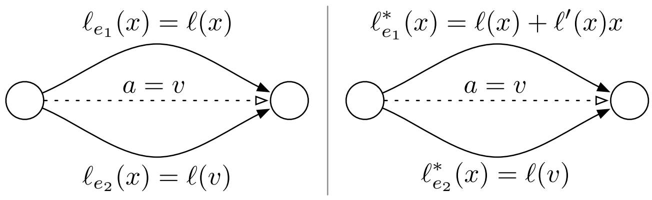
3.4. Der Anarchie-Wert einer Menge von Latenzfunktionen

Roughgarden: Sei $\lambda \in [0,1]$ Lösung für $\ell^*(\lambda v) = \ell(\lambda v) + \ell'(\lambda v)\lambda v = \ell(v)$. Ein solches λ gibt es nach Zwischenwertsatz aufgrund der Stetigkeit von ℓ^*. Dann:

$$\alpha(\ell) := \sup_{\substack{v>0 \\ \ell(v)>0}} \left(\lambda \cdot \frac{\ell(\lambda v)}{\ell(v)} + (1-\lambda) \right)^{-1}$$

$$= \left[1 - \sup_{\substack{v>0 \\ \ell(v)>0}} \lambda \left(\frac{\ell(v) - \ell(\lambda v)}{\ell(v)} \right) \right]^{-1}$$

Erklärung Anarchie-Wert $\alpha(\ell)$: Verhältnis zwischen schlechtestem BE und dem Systemoptimum in folgendem Netzwerk (links):



Ein Systemoptimum im linken Netzwerk ist dabei ein BE im rechten.

Lemma: Sei ℓ diff'bare Latenzfunktion, für die ferner gilt, dass $\ell \cdot \mathrm{id}$ konvex ist. Dann ist $\alpha(\ell) = (1 - \beta(\ell))^{-1}$.

4. Berechnung des höchstmöglichen Preises der Anarchie

Lemma: Sei $\mathcal{L}$ Menge von Latenzfunktionen. Es gebe $s : \mathbb{R} \to \mathbb{R}$ mit $\ell(cx) \geq s(c)\ell(x)$ für alle $\ell \in \mathcal{L}$, $c \in [0,1]$ und $x \in \mathbb{R}_{\geq 0}$. Dann folgt $\beta(\mathcal{L}) \leq \sup_{0 \leq x \leq 1}\{x(1 - s(x))\}$.

Lemma: Es gebe $s : \mathbb{R} \to \mathbb{R}$ mit $\ell(cx) \geq s(c) + \ell(x)$ für alle $\ell \in \mathcal{L}$, $c \in [0,1]$ und $x \in \mathbb{R}_{\geq 0}$. Ferner sei $f \in S$ ein BMW und f^* ein Systemoptimum. Es ist $D := \sum_{i=1}^n a_i$ der Gesamtbedarf. Dann gilt $C(f) \leq C(f^*) - |A| \cdot D \cdot \inf_{0 \leq x \leq 1}\{x \cdot s(x)\}$.

Folgerung: Es gelte $\ell(cx) \geq c \cdot \ell(x)$ für alle $\ell \in \mathcal{L}$, $c \in [0,1]$ und $x \in \mathbb{R}_{\geq 0}$. ($\mathcal{L}$ ist also beliebige Teilmenge der konkaven Funktionen geschnitten mit der Menge aller Latenzfunktionen.) Dann folgt $\alpha(\mathcal{L}) \leq \frac{4}{3}$.

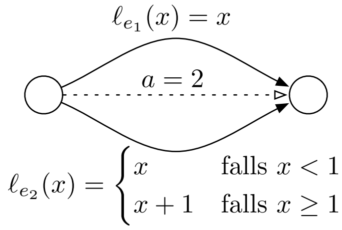
Folgerung: Sei $n \in \mathbb{N}$. Es gelte $\ell(cx) \geq c^n \cdot \ell(x)$ für alle $\ell \in \mathcal{L}$, $c \in [0,1]$ und $x \in \mathbb{R}_{\geq 0}$. Dann folgt

$$\alpha(\mathcal{L}) \leq \frac{(n+1)^{1+1/n}}{(n+1)^{1+1/n} - n}.$$

Folgerung: Sei $n \in \mathbb{N}$. Es gelte $\ell(cx) \geq \log_b(c) + \ell(x)$ für alle $\ell \in \mathcal{L}$, $c \in [0,1]$ und $x \in \mathbb{R}_{\geq 0}$. Es sei $f \in S$ ein BMW und f^* ein Systemoptimum. Es ist $D := \sum_{i=1}^n a_i$ der Gesamtbedarf. Dann gilt $C(f) \leq C(f^*) + \frac{|A| \cdot D}{e \ln b}$.

5. Unstetige Latenzfunktionen

Folgendes Netzwerk mit einer nur rechtsseitig stetigen Latenzfunktion hat kein BE:



Ein BMW existiert aber weiterhin!

5.1. Linksstetige Latenzfunktionen

Lemma: Bei linksseitig stetigen Latenzfunktionen existiert weiterhin ein BMW.

Satz: Die Definition von $\beta(\mathcal{L})$ kann auch für allgemeine Mengen linksstetiger Latenzfunktionen erweitert werden. Es gilt dann weiterhin die Schranke von $(1 - \beta(\mathcal{L}))^{-1}$ für den Preis der Anarchie in Netzwerken mit Latenzfunktionen aus $\mathcal{L}$.

5.2. Allgemeine unstetige Latenzfunktionen

Lemma: Eine sinnvolle Erweiterung von $\beta(\mathcal{L})$ ist nicht möglich.