

## **Zusammenfassung**

In dieser Arbeit zeigen wir, dass es sich bei dem COALITION SKILL GAME (CSG) um ein superadditives Spiel handelt. Bei einem superadditives Spiel ist die große Koalition zwangsläufig die beste Strategie, um den Gewinn aller Agenten zu maximieren. Um diesen Zusammenhang aufzuzeigen, formalisieren wir zunächst das in der Aufgabenstellung beschriebene Spiel: So können wir zeigen, dass jedes CSG bezüglich seiner Strategien equivalent ist zu einem Spiel, in dem jeder Agent nur über eine Skillkapazität verfügt. Anschließend zeigen wir die Superadditivitätseigenschaft für das Spiel und geben ein Algorithmus an, der die optimale Ressourcenverteilung aus der Sicht der Agenten berechnet.

# AOTidf Bagger Game

Denis Erfurt, Tobias Behrens

30. November 2016

## 1 Voraussetzungen

Um das gegebene Problem strukturell analysieren zu können, benutzen wir die Sprache der HOL (Higher Order Logic). Zunächst werden wir in (1.2) vier Signaturen mit den dazugehörigen Modellklassen einführen, auf denen das Problem analysiert wird. Anschließend werden wir in (1.3) Beziehungen zwischen den Modellklassen betrachten und schließlich in (1.4) die Superadditivität des Spiels zeigen.

**TODO:** Vereinfachungen - fixes binäres budget

### 1.1 Grundlagen

**TODO:** HOL

Im weiteren Verlauf der Arbeit werden Funktionen mit kleinem Anfangsbuchstaben und Relationen mit einem großen Anfangsbuchstaben geschrieben. Ebenfalls gehen wir davon aus, dass alle Funktionen total und mit 0 initialisiert sind, falls für eine Struktur Eingabeparameter nicht näher definiert wurden. Wir verwenden die Schreibweise in Prädikatenlogik und die Mengenschreibweise equivalent:  $a \in A \equiv A(a)$

## 1.2 Terminologie

### 1.2.1 Signaturen und Modellklassen

Wir führen nun vier verschiedene Signaturen mit den dazugehörigen Modellklassen ein:

1. **CSGS** - Coalition Skill Game Setting
2. **CSG** - Coalition Skill Game
3. **SCSGS** - Simple Coalition Skill Game Setting
4. **SCSG** - Simple Coalition Skill Game

**Definition 1** (CSGS):

Eine COALITIONAL SKILL GAME SETTING-Signatur (CSGS-Signatur) sei definiert als

$$\sigma_{CSGS} := \{Agent_{/1}, Baustelle_{/1}, supply_{/2}, demand_{/2}, budget_{/1}, kosten_{/4}\}$$

**Intuition**

$Agent(x)$	$:\Leftrightarrow$	$x$ ist ein Agent (Baufirma)
$Baustelle(x)$	$:\Leftrightarrow$	$x$ ist eine Baustelle
$supply(x, t) \mapsto n$	$:\Leftrightarrow$	Agent $x$ besitzt $n$ Einheiten vom Skilltyp $t$
$demand(x, t) \mapsto n$	$:\Leftrightarrow$	Baustelle $x$ benötigt $n$ Einheiten vom Skilltyp $t$
$budget(x) \mapsto n$	$:\Leftrightarrow$	Baustelle $x$ ist bereit einen Gewinn von maximal $n$ bei Fertigstellung auszusahlen
$kosten(t, n, x, y) \mapsto n$	$:\Leftrightarrow$	Kosten für Agenten $x$ für den Transport von $n$ Einheiten des Skilltyp $t$ an die Baustelle $y$ .

Im folgenden werden wir die  $\sigma_{CSGS}$ -Struktur  $\mathcal{S}$  als ein Setting bezeichnen. Die Modellklasse  $M_{CSGS}$  steht für die Gesamtheit an validen Settings. Auf eine formale Definition der Validität eines Settings – z.B. *die Menge der Baustellen und Agenten sind disjunkt* – wird verzichtet.

Beispielhaft ein Setting für ein gegebenes Szenario:

**Beispiel 1:**

Sei  $\mathcal{S}$  eine  $\sigma_{CSGS}$ -Struktur mit:

$Agent^{\mathcal{S}} := \{a_1, a_2, a_3\}$	$Baustelle^{\mathcal{S}} := \{b_1, b_2\}$
$supply^{\mathcal{S}}(a_1, t_1) \mapsto 2$	$demand^{\mathcal{S}}(b_1, t_1) \mapsto 10$
$supply^{\mathcal{S}}(a_1, t_2) \mapsto 7$	$demand^{\mathcal{S}}(b_1, t_2) \mapsto 5$
$supply^{\mathcal{S}}(a_2, t_1) \mapsto 3$	$demand^{\mathcal{S}}(b_2, t_1) \mapsto 2$
$supply^{\mathcal{S}}(a_2, t_2) \mapsto 5$	$demand^{\mathcal{S}}(b_2, t_2) \mapsto 2$
$supply^{\mathcal{S}}(a_3, t_1) \mapsto 20$	$budget^{\mathcal{S}}(b_1) \mapsto 10$
$supply^{\mathcal{S}}(a_3, t_2) \mapsto 5$	$budget^{\mathcal{S}}(b_2) \mapsto 3$

**Definition 2 (CSG):**

Eine COALITIONAL SKILL GAME-Signatur (CSG-Signatur) sei definiert als

$$\sigma_{CSG} := \sigma_{CSGS} \cup \{m_{/3}, v_{/2}\}$$

**Intuition**

$m(x, t, y) \mapsto n \quad :\Leftrightarrow \quad \text{Agent } x \text{ sendet } n \text{ Einheiten des Skilltyps } t \text{ an}$   
die Baustelle  $y$

$v(x, y) \mapsto n \quad :\Leftrightarrow \quad \text{Agent } x \text{ erhält von Baustelle } y \text{ die Vergütung } n$

Wir bezeichnen eine  $\sigma_{CSG}$ -Struktur als ein Game  $\mathcal{G}$ . Intuitiv ist ein Game ein valides Setting mit einer möglichen validen Lösung: eine mögliche Verteilung der verfügbaren Skills der Baufirmen mit entsprechender Vergütungen der Baustellen.

$M_{CSG}$  bezeichnen wir als Klasse aller valider Games. Hier wird ebenfalls auf eine formale Definition valider Kriterien verzichtet. Informell seien aber die wichtigsten Aspekte der Validität kurz beschrieben: Ein Agent kann nicht mehr Einheiten eines Skills ausgeben als er besitzt; Eine Baustelle kann nicht mehr Geld ausgeben, als sie besitzt; Eine Baustelle gibt nur Geld aus, falls sie vollständig mit den benötigten Skills versorgt wird.

### Beispiel 2:

Sei  $\mathcal{G}$  eine Erweiterung der  $\sigma_{CSG}$ -Struktur  $\mathcal{S}$  mit:

$$\begin{array}{l|l} m(a_1, t_1, b_1) \mapsto 2 & m(a_1, t_2, b_1) \mapsto 5 \\ m(a_2, t_1, b_1) \mapsto 3 & m(a_3, t_1, b_1) \mapsto 5 \\ m(a_3, t_1, b_2) \mapsto 2 & m(a_3, t_2, b_2) \mapsto 2 \end{array}$$

**TODO:** vergütung und veranschaulichung

Wir zeigen später, dass sich jedes Setting bzw. Game zu einem Setting bzw. Game vereinfachen lässt, in dem jeder Agent nur eine Skillkapazität besitzt. Diese Vereinfachung erleichtert uns den Beweis der Superadditivität und die Betrachtung möglicher Algorithmen zur Verteilungsberechnung. Hierfür benötigen wir jedoch die neue Relation  $AgentOwner_{/2}$ , mit der wir uns die Zuteilung einer Skillkapazität zu seinem ursprünglichen Agenten merken.

### Definition 3 (SCSGS):

Eine SIMPLE COALITION SKILL GAME SETTING-Signatur sei definiert als

$$\sigma_{SCSGS} :=$$

$$\{agentOwner_{/2}, Agent_{/1}, Baustelle_{/1}, supply_{/2}, demand_{/2}, budget_{/1}, kosten_{/4}\}$$

### Intuition

$AgentOwner(a, x)$	$:\Leftrightarrow$	Agent $x$ gehört zu dem ursprüngl. Agenten $a$
$Agent(x)$	$:\Leftrightarrow$	$x$ ist ein Agent (Baufirma)
$Baustelle(x)$	$:\Leftrightarrow$	$x$ ist eine Baustelle
$supply(x, t) \mapsto n$	$:\Leftrightarrow$	Agent $x$ besitzt $n$ Einheiten vom Skilltyp $t$
$demand(x, t) \mapsto n$	$:\Leftrightarrow$	Baustelle $x$ benötigt $n$ Einheiten vom Skilltyp $t$
$budget(x) \mapsto n$	$:\Leftrightarrow$	Baustelle $x$ ist bereit einen Gewinn von maximal $n$ bei Fertigstellung auszus zahlen
$kosten(t, n, x, y) \mapsto n$	$:\Leftrightarrow$	Kosten für Agenten $x$ für den Transport von $n$ Einheiten des Skilltyp $t$ an die Baustelle $y$ .

Wir bezeichnen eine  $\sigma_{SCSGS}$ -Struktur  $\mathcal{SS}$  als *Simple Setting* und die dazugehörige Modellklasse valider Strukturen  $M_{SCSGS}$ . Auch hier werden wir die Validitätskriterien nicht weiter erläutern.

**Definition 4 (SCSG):**

Eine SIMPLE COALITION SKILL GAME-Signatur (SCSG) sei definiert als

$$\sigma_{SCSG} := \sigma_{SCSGS} \cup \{M_{/3}, v_{/2}\}$$

**Intuition**

$m(x, t, y) \mapsto n \quad :\Leftrightarrow \quad$  Agent  $x$  sendet  $n$  Einheiten des Skilltyps  $t$  an  
die Baustelle  $y$

$v(x, y) \mapsto n \quad :\Leftrightarrow \quad$  Agent  $x$  erhält von Baustelle  $y$  die Vergütung  $n$

Wir bezeichnen eine  $\sigma_{SCSG}$ -Struktur  $\mathcal{SG}$  als *Simple Game* mit der dazugehörigen validen Modelklasse  $M_{SCSG}$ . Validitätskriterien werden an dieser Stelle nicht weiter betrachtet.

**1.3 bez**

Bei der gegebenen Aufgabenstellung interessieren wir uns für ein Mechanismus der bei einem gegebenen Setting ein Game berechnet. Formal gesehen:

$$m : M_{CSGS} \leftarrow M_{CSG} \tag{1}$$

Nach welchen Kriterien dieser Mechanismus arbeitet und bewertet wird wird vorerst offen gelassen.

Die nachfolgenden Betrachtungen wollen wir jedoch auf einer vereinfachten Strukturen machen, bei denen jeder Agent nur eine Ressource besitzt. Um dennoch Aussagen über CSG machen zu können werden wir zeigen dass jedes Setting bzw. Game zu einem Simple Setting bzw. Simple Game überführen lässt. Dieses erlaubt uns auch die Betrachtung von Mechanismen die bei einem Simple Setting ein Simple Game berechnen. Formal:

$$\begin{array}{ccc}
M_{CSGS} & \xrightarrow{\pi'^{-1} \circ m' \circ \pi} & M_{CSG} \\
\downarrow \pi & & \uparrow \pi'^{-1} \\
M_{SCSGS} & \xrightarrow{m'} & M_{SCSG}
\end{array}$$

Dabei müssen folgende Eigenschaften gelten:

$$\pi \quad - \quad \text{total, injektiv} \quad (2)$$

$$\pi^{-1} \quad - \quad \text{surjektiv} \quad (3)$$

$$\pi' \quad - \quad \text{total, injektiv} \quad (4)$$

$$\pi'^{-1} \quad - \quad \text{surjektiv} \quad (5)$$

$$\pi^{-1} \circ \pi \quad = \quad id_{M_{CSGS}} \quad (6)$$

$$\pi'^{-1} \circ \pi' \quad = \quad id_{M_{CSG}} \quad (7)$$

$$(8)$$

Weiter werden wir nur Mechanismen betrachten die nichts am Setting ändern sondern nur matchings und vergütungsverteilung berechnen. Auf eine ausführliche Definition der gesuchten Funktionen wird hier ebenfalls verzichtet. Stattdessen wird eine intuition gegeben: Ein Setting bzw. Game lässt sich in ein Simple Setting bzw. Simple Game überführen indem jede resource eines Agenten als eigenständiger Agent betrachtet wird wobei die zugehörigkeit von agenten unr Ressourcen sich in der *AgentOwner* Relation gemerkt wird. Kostenfunktion bleibt dabei bestehen und wird lediglich auf die Agentbesitzer übertragen. Bei den Matches sowie vergütung wird ähnlich verfahren. Hierdurch gehen keine informationen verlohren und alle Betrachtungen können auf den vereifachten Modellen durchgeführt werden.

## 1.4 supadd

Um die Superadditivität des Spieles zu zeigen werden zunächst der Begriffe formal festgehalten:

**Definition 5:**

Sei  $K \subseteq A$  eine beliebige Koalition. Wir definieren die Menge  $PM(K)$  der potentiellen Matches:

$$PM(K) := \{ \quad (9)$$

$$m : Agent \times Resource \times Baustelle \rightarrow \mathbb{N} \mid (10)$$

$$\phi_{Match}(m) \wedge (11)$$

$$\forall x. \forall t. \forall y. m(x, t, y) > 0 \rightarrow x \in K \} (12)$$

Dabei definiert  $\phi_{Match}(m)$  die validen Matches:

$$\phi_{Match}(m) := \forall x. \forall t. \left( \sum_{y \in Baustelle} m(x, t, y) \right) \leq supply(x, t) \quad (13)$$

Die Zeile (12) sagt das nur solche Matches betrachtet werden, deren "aktive" Agenten auch teil der Koalition sind.

**Definition 6:**

Sei  $K$  eine beliebige Koalition, dann definiert  $B(K)$  die Menge der möglichen Baustellen, die von  $K$  gleichzeitig gebaut werden können.

$$B(K) := \{ B_k \subseteq Baustelle \mid (14)$$

$$\exists m \in PM(K). \forall b \in B_k. \forall t. \left( \sum_{x \in K} m(x, t, b) \right) \geq demand(b, t) \} (15)$$

Da es sich um ein Simple Game handelt, Könnenn wir o.B.d.A. annehmen, dass eine Agent nur einer Koalition zugeordnet ist:

$$K \neq S \Rightarrow K \cap S = \emptyset \quad (16)$$

Hieraus folgt notwendigerweise, dass eine Baustelle in einem Spiel nur von einer Koalition gebaut werden kann, da diese pro Spiel nur einmal gebaut werden kann. Um dieses formal festzuhalten definieren wir eine Outcome Relation:



**Definition 7:**

Seien  $K, S$  Koalitionen. Eine Outcome Menge weist beiden Koalitionen ein Tupel der Baustellen zu, die sie jeweils zur gleichen Zeit bauen können:

$$Outcome(K, S) := \{(B_K, B_S) \subseteq Baustelle \times Baustelle \mid \quad (17)$$

$$B_K \in B(K) \wedge \quad (18)$$

$$B_S \in B(S) \wedge \quad (19)$$

$$B_K \cap B_S = \emptyset\} \quad (20)$$

Zur weiteren vereinfachung definieren wir uns noch eine Bewertungsfunktion für eine Menge von Baustellen:

**Definition 8:**

Sei  $B \subseteq Baustelle$  beliebig. Dann definiert sich der wert von B:

$$v(B) := \sum_{b \in B} budget(b) \quad (21)$$

**Definition 9:**

Nun können wir den Wert betrachten, die zwei Koalitionen erzielen können. Seien  $K, S$  beliebige Koalition:

$$v(K) + v(S) := \quad (22)$$

$$\max_{(B_K, B_S) \in Outcome(K, S)} (v(B_K) - \varphi_{kosten}(K, B_K) + v(B_S) - \varphi_{kosten}(S, B_S)) \quad (23)$$

Dadurch dass die Baustellen die von einer Koalition gebaut werden nicht mehr von einer anderen koalition gebaut werden können, müssen wir diese zeitgleich betrachten.

$\varphi_{kosten}(B, K)$  ist dabei die funktion, die basierend auf der vorhandenen Kostenfunktion, der Koalition und den Baustellen die kosten für die Koalition berechnet alle baustellen fertig zu stellen. **Bemerkung:** Nach Definition ist das Ergebniss pareto effizient.

Exestiert eine Koalition isoliert, lässt sich diese auch isoliert betrachten:

**Definition 10:**

Sei  $K$  eine beliebige Koalition.

$$v(K) := \max_{B' \in B(K)} (v(B') - \varphi_{kosten}(B', K)) \quad (24)$$

Insbesondere gilt dann auch für zwei allein agierende Koalitionen  $K, S$ :

$$v(K \cup S) = \max_{B' \in B(K \cup S)} (v(B') - \varphi_{kosten}(B', K)) \quad (25)$$

Beachte auch das für einen Outcome zweier Koalitionen eine mögliche Baustellenmenge der potentiellen Baustellen der vereinigung der Koalitionen exestiert die alle Baustellen des seperaten Outcomes beinhaltet:

$$\forall (B_K, B_S) \in Outcome(K, S). \exists B' \in B(K \cup S). B_K \cup B_S \subseteq B' \quad (26)$$

Hieraus können wir die Superadditivität des Spiels schließen:

$$v(K \cup S) \geq v(K) + v(S) \quad (27)$$

Insbesondere ist das Spiel Konvex. Nach einem Theorem (TODO) ist bei einem konvexen Spiel das Shapley Value im Core enthalten, sodass die beste Lösung dieses Spiels die große Koalition ist und eine stabile und faire Lösung immer exestiert.