AOTidf Bagger Game

Denis Erfurt, Tobias Behrens, Abdallah Kadour

ABSTRACT

TODO: Abstract

1 GLIEDERUNG

TODO: gliederung

2 VORAUSSETZUNGEN

In diesem Kapitel geben wir in Abschnitt (2.1) einen Gesamtüberblick über die Aufgabenstellung und formulieren Prämissen für die weitere Bearbeitung. Im Abschnitt (2.2) werden notwendige theoretische Grundlagen zusammengefasst, die für das weitere Verständnis der Arbeit notwendig sind.

2.1 Aufgabenstellung

Das gegebene Szenario lässt sich wie folgt zusammenfassen: Es existieren eine Menge an Baustellen und Baufirmen, im Folgenden Agenten genannt. Agenten verfügen über ein Bestand an Skillkapazitäten – eine Anzahl an Einheiten verschiedener Skills – und Baustellen wiederum benötigen Skillkapazitäten für ihre Fertigstellung. Wird die Baustelle fertiggestellt, wird eine vorher definierte Summe – der Erlös – ausgeschüttet.

Diese Arbeit modelliert dieses Szenario (Kapitel 3) und untersucht Mechanismen (Kapitel 4), die mit Hilfe von Koalitionsbildung eine Zuordnung von Bestand und Bedarf an Skillkapazitäten bestimmen. Die Mechanismen werden bezüglich der Maximierung der sozialen Wohlfahrt, Stabilität der gebildeten Koalitionen und Fairness der Gewinnausschüttung, sowie auch ihrer absoluten Performance untersucht.

Unter **sozialer Wohlfahrt** verstehen wir die Summe der Gewinne aller Agenten. Der Gewinn eines Agenten a ist dabei die Differenz zwischen dem Teil des Erlöses, den a aus dem Gesamterlös erhält, und den Kosten, die für a im Spiel für die Bereitstellung seiner Skillkapazitäten tatsächlich anfallen.

Ein Mechanismus ist in unserem Verständnis bezüglich seiner Gewinnausschüttung **fair**, wenn dessen Gewinnausschüttung für jeden Agenten dem *Shapley Value* des Agenten entspricht. Wir folgen mit unserem Fairness-Begriff also dem normativen Verständnis des Shapley Values.[1]

[1] Auf eine formale Definition sei an dieser Stelle verzichtet und unter anderem auf [4] verwiesen, wo auch vielfältige weitere Anwendungen des Shapley Values aufgezeigt werden.

Der Mechanismus soll weiterhin bezüglich seiner Koalitionsbildung **stabil** sein. Das heißt, dass kein Agent einer durch den Mechanismus gebildeten Koalition einen Anreiz hat, diese Koalition zu verlassen, um seinen Gewinn alleine oder durch die Bildung einer neue Koalition zu maximieren. Diese Koalitionen befindet sich im *Core* [2].

Prämissen. Des Weiteren gehen wir von folgenden Grundannahmen bei der Bearbeitung der Aufgabenstellung aus:

(1) **Rationalität**: Agenten arbeiten ausschließlich für ihr eigenes Interesse.
(2) **Multiskill**: Agenten können mehrere Skilltypen mit beliebiger Quantität besitzen.
(3) **Linearität**: Skillkapazitäten können höchstens ein mal eingesetzt werden und sind nach ihrem Einsatz ”verbraucht”.
(4) **unvollständige Information der Konkurrenz**: Agenten haben keine Information über die Ressourcen anderer Agenten und treten gegeneinander in Konkurrenz.
(5) **vollständige Informationen des Bedarfs**: Agenten haben vollständige Information über die Anzahl, Position, Bedarf sowie Vergütung der Bauaufträge.
(6) **Zeitagnostisch**: Alle Betrachtungen werden ohne Berücksichtigung der Zeit gemacht. Insbesondere erfolgen über die Zeit keine Änderungen an den Bauaufträgen oder den Skillkapazitäten Agenten.

2.2 Grundlagen

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 werden. Wir verwenden die Schreibweise in Prädikatenlogik und die Mengenschreibweise äquivalent: $a \in A \equiv A(a)$ Um das gegebene Problem strukturell analysieren zu können, benutzen wir die Sprache der HOL (Higher Order Logic).

3 MODELLIERUNG

In diesem Kapitel verfolgen wir das Ziel, durch eine Formalisierung der Aufgabenstellung eine geeignete Grundlage zu für die Formulierung von Mechanismen und die Beantwortung zentraler Fragestellungen zu schaffen. Zunächst werden wir in Abschnitt

(3.2) grundlegende Definitionen vorstellen sowie in Abschnit (3.1) zwei Signaturen sowie die dazugehörigen Modellklassen einführen.

3.1 Signaturen und Modellklassen

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

(1) **CSGS** - Coalition Skill Game Setting
(2) **CSG** - Coalition Skill Game

Definition 3.1 (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

$Agent(x)$	:⇔	x ist ein Agent (Baufirma)
$Baustelle(x)$	:⇔	x ist eine Baustelle
$supply(x, t) \mapsto n$	:⇔	Agent x besitzt n Einheiten vom Skilltyp t
$demand(x, t) \mapsto n$	:⇔	Baustelle x benötigt n Einheiten vom Skilltyp t
$budget(x) \mapsto n$	:⇔	Baustelle x zahlt einen Gewinn n bei Fertigstellung aus
$kosten(t, n, x, y) \mapsto n$	:⇔	Kosten für Agenten x für die Bereitstellung von n Einheiten des Skilltyp t an Baustelle y.

Im folgenden werden wir die σ_{CSGS}-Struktur $\mathcal{S}$ als ein Setting bezeichnen. Die Modellklasse M_{CSGS} steht für die Gesamtheit an validen Settings, die zudem die folgende Bedingung erfüllt:

$$Agent \cap Baustelle = \emptyset \tag{1}$$

Beispiel 1:
Ein Beispiel für eine valides Setting ist die Struktur $\mathcal{S} \in M_{CSGS}$ 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 3.2 (CSG). Eine Coalitional Skill Game-Signatur (CSG-Signatur) sei definiert als

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

Intuition

$m(x, t, y) \mapsto n$	:⇔	Agent x sendet n Einheiten des Skilltyps t an die Baustelle y
$v(x, y) \mapsto n$	:⇔	Agent x erhält von Baustelle y die Vergütung n

Wir bezeichnen eine σ_{CSG}-Struktur als ein Game $\mathcal{G}$. Ein Game ist also ein valides Setting mit einer validen Lösung: eine mögliche Zuordnung verfügbarer Skills von Baufirmen zu Baustellen, die zu einer definierten Erlösausschüttung der Baustellen führt.
M_{CSG} bezeichnen wir als Klasse aller valider Games, die folgende (informale) Eigenschaften erfüllen:

(1) Ein Agent kann nicht mehr Einheiten eines Skills ausgeben als er besitzt.
(2) Eine Baustelle schüttet genau dann ihr Geld aus, wenn sie vollständig mit den benötigten Skills versorgt wird.

Sei $M_{OCSG} \subset M_{CSG}$ eine Modellklasse die zusätzlich "optimal" ist: Es existiert kein Matching, bei dem der summierte Gewinn über alle Agenten größer ist. Sie enthält so alle Zuordnungen, die die soziale Wohlfahrt maximieren.
Sei $M_{FOCSG} \subset M_{OCSG}$ eine Modellklasse, die zusätzlich "fair" ist: Die Erlösverteilung an die Agenten entspricht damit in unserem Verständnis dem Shapley-Value: Jeder Agent erhält den Durchschnitt seiner marginalen Beiträge zu jeder möglichen Permutation der Reihenfolge der Spieler. Der marginale Beitrag eines Spielers a zu einer Reihenfolge (betrachtet als Koalition[2]) ist definiert als Differenz des Gewinns zwischen der Koalition aus allen Agenten vor a in der Reihenfolge und der Koalition aus allen Agenten vor a und mit a in der Reihenfolge.
Insbesondere interessieren wir uns hier für die Frage, ob eine faire Verteilung existiert: $M_{FOCSG} =^? \emptyset$

3.2 Terminologie

Definition 3.3 (Koalition). Eine **Koalition** ist ein Zusammenschluss von Agenten, die vollständige Informationen über alle Koalitionsteilnehmer besitzen und die Verteilung gemeinsamer Ressourcen auf Bauaufträge beabsichtigen. Ebenfalls haben sie einen gemeinsamen Mechanismus, welcher den Erlös auf die Koalitionsteilnehmer aufteilt. Formal: $K \subseteq Agenten$ mit folgenden Eigenschaften:

$$\exists x \exists t \exists y. m(x, t, y) > 0 \wedge K(x) \rightarrow (\forall x' \forall t' m(x', t', y) > 0 \Rightarrow K(x')) \tag{2}$$

$$\exists x \exists y v(x, y) > 0 \wedge K(x) \rightarrow (\forall x'. v(x', y) > 0 \rightarrow K(x')) \tag{3}$$

Dabei sagt Bedingung (2) aus, dass eine Baustelle nur von einer Koalition gebaut werden kann und Bedingung 3, dass eine Baustelle nur Geld an eine Koalition senden kann.

[2] näher definiert im nächsten Abschnitt

Definition 3.4 (Matching). Ein **Matching** ist eine konkrete Zuordnung von Skillkapazitäten zu Bauaufträgen. Formal das Prädikat $m \in \sigma_{CSG}$.

3.3 Mechanismen

TODO: Mechanismuskriterien

TODO: Phasen

TODO: überarbeiten

Bei der gegebenen Aufgabenstellung interessieren wir uns für ein Mechanismus, der bei einem gegebenen Setting ein optimales Game mit einer fairen Verteilung dezentral hervorbringt. Formal gesehen:

$$m : M_{CSGS} \to M_{FOCSG} \tag{4}$$

4 ERGEBNISSE

4.1 Theoretische Ergebnisse

4.1.1 Superadditivität. Im Folgenden zeigen wir die Superadditivität das CSG. Intuitiv zeigen wir, dass ein Spieler immer nur einen positiven Wert in eine Koalition einbringt.

Lemma 4.1 (Superadditivität). *Das CSG ist superadditiv.*

Bevor wird die Superadditivität des CSG zeigen, führen wir zunächst zwei weitere "simple" Signaturen mit dazugehörigen Modellklassen ein und zeigen, dass Ergebnisse auf ihnen direkt auf das CSG übertragen werden können. Außerdem führen wir die Begriffe *potentielle Matches* einer Koalition, die Menge der *gleichzeitig möglichen Baustellen einer Koalition*, die *Outcome-Menge* von zwei Koalitionen und den *Wert* einer Baustelle ein.

(1) **SCSGS** - Simple Coalition Skill Game Setting
(2) **SCSG** - Simple Coalition Skill Game

Wir zeigen im folgenden, dass sich jedes Setting bzw. Game zu einem Setting bzw. Game vereinfachen lässt, in dem jeder Agent nur eine Einheit eines Skilltypen 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 4.2 (SCSGS). Eine Simple Coalition Skill Game Setting-Signatur sei definiert als

$$\begin{aligned}\sigma_{SCSGS} :=& \{gentOwner_2, Agent_{/1}, Baustelle_{/1}, supply_{/2}, \\ & demand_{/2}, budget_{/1}, kosten_{/4}\}\end{aligned}$$

Intuition

$AgentOwner(a, x)$	$:\Leftrightarrow$	Agent x gehört zu dem Agenten a
$Agent(x)$	$:\Leftrightarrow$	x ist ein Agent (Baufirma)
$Baustelle(x)$	$:\Leftrightarrow$	x ist eine Baustelle
$type(x) \mapsto t$	$:\Leftrightarrow$	Agent x ist vom Skilltyp t
$demand(x, t) \mapsto n$	$:\Leftrightarrow$	Baustelle x benötigt n Einheiten vom Skilltyp t
$budget(x) \mapsto n$	$:\Leftrightarrow$	Baustelle x zahlt einen Gewinn n bei Fertigstellung aus
$kosten(t, n, a, y) \mapsto n$	$:\Leftrightarrow$	Kosten für zugehörigen Agenten a für die Bereitstellung von n Einheiten des Skilltyps t an Baustelle y

Wir bezeichnen eine σ_{SCSGS}-Struktur $\mathcal{SS}$ als *Simple Setting* und die dazugehörige Modellklasse valider Strukturen M_{SCSGS}. Validitätskritärien sind analog zu M_{CSGS}.

Definition 4.3 (SCSG). Eine Simple Coalition Skill Game-Signatur (SCSG) sei definiert als

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

Intuition

$m(x, t, y)$	$:\Leftrightarrow$	Agent x sendet eine Einheiten des Skilltyps t an die Baustelle y
$v(x, y) \mapsto n$	$:\Leftrightarrow$	Agent x erhält von Baustelle y die Vergütung n

Wir bezeichnen eine σ_{SCSG}-Struktur $\mathcal{SG}$ als *Simple Game* mit der dazugehörigen validen Modelklasse M_{SCSG}. Validitätskriterien sing analog zu M_{CSG}.

4.2 Beziehungen zwischen den Modellklassen

Die nachfolgenden Betrachtungen wollen wir jedoch auf der vereinfachten Strukturen anstellen: Bei dieser besitzt jeder Agent nur eine Einheit eines Skilltyps. Um dennoch Aussagen über das CSG machen zu können, zeigen wir, dass sich jedes Setting bzw. Game zu einem Simple Setting bzw. Simple Game überführen lässt. Das erlaubt uns die Ergebnisse von theoretische Betrachtungen von Mechanismen, die bei einem Simple Setting ein Simple Game berechnen, auf das CSG zu übetragen.
Formal:

$$\begin{tikzcd}
M_{CSGS} \arrow[r, "\pi'^{-1} \circ m' \circ \pi"] \arrow[d, "\pi"] & M_{CSG} \\
M_{SCSGS} \arrow[r, "m'"] & M_{SCSG} \arrow[u, "\pi'^{-1}"']
\end{tikzcd}$$

Dabei müssen folgende Eigenschaften gelten:

$$\pi \quad - \quad \text{total, injektiv} \tag{5}$$
$$\pi^{-1} \quad - \quad \text{surjektiv} \tag{6}$$
$$\pi' \quad - \quad \text{total, injektiv} \tag{7}$$
$$\pi'^{-1} \quad - \quad \text{surjektiv} \tag{8}$$
$$\pi^{-1} \circ \pi \quad = \quad id_{M_{CSGS}} \tag{9}$$
$$\pi'^{-1} \circ \pi' \quad = \quad id_{M_{CSG}} \tag{10}$$

Weiter werden wir nur Mechanismen betrachten, die am Setting keine Änderungen vornehmen, sondern nur Matchings und die Vergütungsverteilung bestimmen. 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 Einheit eines Skilltyps eines Agenten als eigenständiger Agent betrachtet wird.
Dabei wird die Zugehörigkeit von Agenten zum Skill in der *AgentOwner* Relation gesichert. Die Kostenfunktion bleibt bestehen und wird lediglich auf den *AgentOwner* übertragen. Bei den Matches und Vergütungen wird analog verfahren.

Definition 4.4 (Potentielle Matches einer Koalition). Sei $K \subseteq Agent$ eine beliebige Koalition. Für K definieren wir die *Menge der potentiellen Matches* $PM(K)$ als:

$$PM(K) := \{ \quad m : Agent \times Skilltyp \times Baustelle \to \mathbb{N} \mid \tag{11}$$
$$\phi_{Match}(m) \wedge \tag{12}$$
$$\forall x\ \forall t\ \forall y : m(x,t,y) > 0 \to x \in K \quad \} \tag{13}$$

Dabei bezeichnet $\phi_{Match}(m)$ die Menge der validen Matches:

$$\phi_{Match}(m) := \forall x\ \forall t\ : \left(\sum_{y \in Baustelle} m(x,t,y) \right) \leq supply(x,t) \tag{14}$$

Der Ausdruck (13) verdeutlicht, dass nur solche Matches für eine Koalition betrachtet werden, deren liefernder Agent auch Teil der Koalition ist.

Definition 4.5 (Gleichzeitig mögliche Baustellen einer Koalition). Sei $K \subseteq Agent$ eine beliebige Koalition. Die Menge der möglichen Baustellen, die von K gleichzeitig gebaut werden können, $B(K)$, ist definiert als:

$$B(K) := \{ \quad B_k \subseteq Baustelle \mid \tag{15}$$
$$\exists m \in PM(K),\ \forall b \in B_k,\ \forall t : \tag{16}$$
$$\left(\sum_{x \in K} m(x,t,b) \right) \geq demand(b,t) \quad \} \tag{17}$$

Da wir ein Simple Game betrachten, Können wir o.B.d.A. annehmen, dass eine Agent nur einer Koalition zugeordnet ist:

$$K \neq S \Rightarrow K \cap S = \emptyset \tag{18}$$

Hieraus folgt, dass eine Baustelle in einem Spiel nur von einer Koalition gebaut werden kann, da eine Baustelle pro Spiel nur einmal gebaut wird. Um dies formal festzuhalten, definieren wir im Folgenden die Outcome-Relation:

Definition 4.6 (Outcome-Menge zweier Koalitionen). Seien $K, S \subseteq Agent$ beliebige Koalitionen. Eine *Outcome-Menge* weist beiden Koalitionen ein Tupel der Baustellen zu, die sie jeweils zur gleichen Zeit bauen können:

$$Outcome(K,S) := \{ \quad (B_K, B_S) \subseteq Baustelle \times Baustelle \mid \tag{19}$$
$$B_K \in B(K) \wedge B_S \in B(S) \wedge B_K \cap B_S = \emptyset \quad \} \tag{20}$$

Beachte auch das für einen Outcome zweier Koalitionen eine mögliche Baustellenmenge der potentiellen Baustellen der Vereinigung der Koalitionen existiert die alle Baustellen des separaten Outcomes beinhaltet:

$$\forall (B_K, B_S) \in Outcome(K,S),\ \exists B' \in B(K \cup S) :\ B_K \cup B_S \subseteq B' \tag{21}$$

Hieraus können wir die Superadditivität des Spiels schließen:

$$v(K \cup S) \geq v(K) + v(S) \tag{22}$$

□

4.2.1 *Instabilität der großen Koalition.*

Lemma 4.7 (Instabilität der grossen Koalition). *Im allgemeinen Fall ist die große Koalition $K = Agenten$ instabil.*

Beweis durch Beispiel:
Im Szenario in der Abbildung (1) ist der Gewinn der Koalitionen $K_1 = \{A0, A1, A2\}$ und $K_2 = \{A0, A2\}$ gleich, jedoch erhält die Agenten $A0$ in K_2 anteilig mehr vom Gewinn. Demnach lohnt es sich für ihn die große Koalition zu verlassen.

□

4.2.2 *NP-Härte des Problems.*

Im folgenden zeigen wir die NP-Härte eines gewünschten Mechanismus. Hierfür reduzieren wir das Rucksackproblem, ein allgemein bekanntes NP-vollständiges Problem [3], auf ein CSG. Das Rucksackproblem ist folgendermaßen definiert:

Lemma 4.8. *Sei $m : M_{CSGS} \to M_{OCSG}$ ein Mechanismus, der ein optimales Matching berechnet, dann ist m NP-hart.*

Definition 4.9 (KNAPSACK). Sei U eine Menge von Objekten, w eine Gewichtsfunktion und v eine Nutzenfunktion:

$$w : U \to \mathbb{R} \tag{23}$$
$$v : U \to \mathbb{R} \tag{24}$$

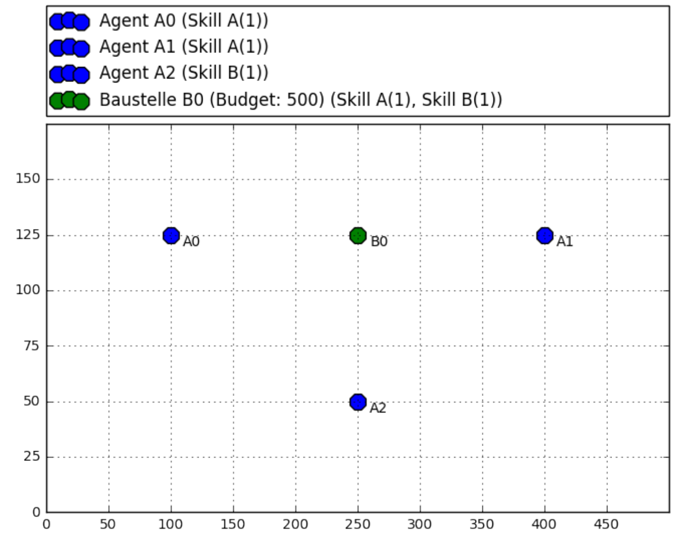

Abbildung 1: Szenario mit einer instabilen großen Koalition.

Sei weiter $B \in \mathbb{R}$ eine vorgegebene Gewichtsschranke. Gesucht ist eine Teilmenge $K \subseteq U$, die folgende Bedingungen erfüllt:

$$\sum_{w \in K} w(u) \leq B \tag{25}$$

$$max(\sum_{w_i n K} v(u)) \tag{26}$$

Sei $K = \{U, w, v\} \in KNAPSACK$ ein beliebiges Rucksackproblem. Wir geben eine Übersetzung in eine CSGS-Signatur an:

$$Agent = \{a\} \tag{27}$$

$$supply(a, t) \mapsto B \tag{28}$$

$$Baustelle = U \tag{29}$$

$$demand(u, t) \mapsto w(u) \tag{30}$$

$$budget(x) \mapsto v(x) \tag{31}$$

$$kosten(t, n, x, y) \mapsto 0 \tag{32}$$

Dabei wird das KNAPSACK-Problem als ein Spiel mit nur einem Agenten und einem Skilltyp verstanden. Die maximale Kapazität des Rucksacks B ist die Quantität dieses Skilltypes des Agenden. Die Menge der Objekte wird als Menge an Bauaufträgen interpretiert, das Gewicht eines Objektes $w(u)$ als die zur Fertigstellung geforderten Ressourcen und dessen Wert $v(u)$ das Auszahlungsbetrages einer Baustelle bei Fertigstellung.
Intuitiv sorgt die Bedingung 25 dafür, dass der Agent seine Ressourcen nicht überschreitet, sowie die Bedingung 26, dass der Agent sein Gewinn maximiert. Beides sind ebenfalls Anforderungen, die von einem Mechanismus erfüllt werden muss, der ein optimales CSG berechnet. Angenommen es gäbe ein $m \in P$ der aus einem CSGS ein CSG mit einem optimalem Matching berechnet, dann wäre auch $KNAPSACK \in P$. Dieses ist jedoch ein Widerspruch zur Annahme $KNAPSACK \in NP$.

□

Stabilisierung. Wie wir in Abschnitt (4.2.1) gesehen haben, ist die große Koalition instabil. Im Folgenden möchten wir in einem Vorschlag zeigen, wie ein Mechanismus zu Koalitionsbildung funktionieren könnte, der eine stabile große Koalition erzeugt: Eine Koalition garantiert ihren Koalitionsteilnehmern beim Beitritt einen individuellen Gewinn in Höhe des Shapley Values und verlangt dafür einen hinreichenden Beitrittsdepot x. Dieser Kollateralbetrag wird von einer unabhängigen Instanz verwaltet und sichert jeden Agenten a vor dem "Rauswurfäus der großen Koalition ab, indem diese Instanz a in der gleichen Höhe auszahlt, wie auch ihr Gewinn in der großen Koalition wäre. Am Schluss wird der Kollateralbetrag wieder anteilig an die Agenten zurück verteilt. Durch wird die große Koalition stabilisiert, da es für Teilkoalitionen nicht mehr lohnt auszusteigen: Der Kollateralbetrag würde um genau den gleichen Wert schrumpfen, wie die Auszahlung der zurückgelassenen Agenten. Der Beitrittsdepot für jeden Agenten für die Große Koalition ist (ohne Beweis)

$$x = \frac{\sum_{b \in Baustellen} budget(b)}{2 * |Agenten|}. \tag{33}$$

Mechanismus. In Abschnitt (4.1.1) sowie (4.2.2) haben wir gezeigt, dass der Beitritt zur großen Koalition die rational richtige Entscheidung für einen Agenten ist, was die Offenlegung aller Skillkapazitäten zur Folge hat. Die Berechnung eines optimalen Matchings ist entscheidbar, jedoch wie wir aus Abschnitt (4.2.2) wissen, NP-hart. Das Berechnen des Shapley-Values ist ebenfalls entscheidbar, jedoch ebenfalls NP-hart[3]. So haben wir in diesem Kapitel gezeigt, dass es ein NP-harten Mechanismus gibt, der bei einem gegebenen Setting ein faires und optimales Ergebnis berechnet, welches den Anforderungen der Aufgabenstellung (2.1) gerecht wird.

4.3 Praktische Ergebnisse

5 ZUSAMMENFASSUNG

TODO: Zusammenfassung

LITERATUR

[1] Vincent Conitzer and Tuomas Sandholm. 2004. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the 19th national conference on Artifical intelligence.* AAAI Press, 219–225.

[2] Donald B Gillies. 2016. SOLUTIONS TO GENERAL NON-ZERO-SUM GAMES1. *Contributions to the Theory of Games (AM-40)* 4 (2016), 47.

[3] Für eine Untersuchung der Komplexität der Berechnung des Shapley Values sei auf [1] verwiesen.

[3] Richard M. Karp. 1972. *Reducibility among Combinatorial Problems.* Springer US, Boston, MA, 85–103. DOI: http://dx.doi.org/10.1007/978-1-4684-2001-2_9

[4] Stefano Moretti and Fioravante Patrone. 2008. Transversality of the Shapley value. *TOP* 16, 1 (2008), 1. DOI: http://dx.doi.org/10.1007/s11750-008-0044-5