

scratch_tobias

November 14, 2016

0.1 Recap

Die erste Idee sah ein Commitment-Verfahren zur Festschreibung der Entlohnung/Kompensation zwischen der Baustelle und der Baufirma bzw. Baufirma und Baufirma bei Koalitionen mit mehreren teilnehmenden Baufirmen vor.

0.2 Neue Ansatz: Reverse Auctions

Eine Baustelle hat eine maximale Summe an Geld, die ausgegeben werden kann. Agenten (Bau-firma) a bietet zu einem geheim generierten Preis $Preis_a(S_{SK_1}(K))$ eine seiner Skillkapazitäten $S_{SK_1}(K)$ vom Skill SK_1 an. Der Payoff eines Agenten bei diesem Preis ist $Payoff_a(S_{SK_1}(K)) = Preis_a(S_{SK_1}(K)) - (K * Preis(SK_1) + Transaktionskosten)$.

Transaktionskosten bestehen aus zwei Bestandteilen: Die Bereitstellungskosten auf Grund-lage der Entfernung zur Baustelle machen neben den Verwaltungskosten den größten Anteil aus. Verwaltungskosten fallen für die Baufirma nur einmal für jede Baustelle an, auch wenn mehrere Skillkapazitäten versteigert werden.

Solange noch keine Koalition gefunden ist, d.h. alle Anforderung an Skillkapazitäten der Baustelle befriedigt sind mit einem Preis, der in der Summe kleiner ist als der Maximalpreis der Baustelle, wird auktioniert. Sobald ein Zustand auftritt, dass kein günstigerer Preis für eine Skillkapazität angeboten wird, wird diese vergeben. Die noch zu vergebenen Skillkapazitäten der Baustelle werden weiter auktioniert.

0.2.1 Derzeitige Vereinfachungen:

- Kommunikation findet nicht zur Absprache statt: Baufirmen verhandeln über Koalitionen indirekt durch Reverse Auction
- rundenbasierte Betrachtung des Spielverlaufs: In fest definierten Phasen werden Angebote ausgeschrieben, Angebote generiert, Ergebnisse der Reverse Auction bekanntgegeben.
- rundenbasierte Betrachtung der Auktionen: keine Berücksichtigung der Zeit bei der Kommunikation, wodurch z.B. kein Vorteil durch schnellere Kommunikation entsteht.
- Transaktionskosten bestehen aus Bereitstellungskosten und Verwaltungskosten: Bereitstellungs-kosten fallen je Skill in definierten Schritten an, Verwaltungskosten je Baustelle.
- Skillkapazitäten werden nicht als Paket angeboten, sondern jede Skillkapazität einzeln

0.2.2 Offene Fragen

- Welche Information bekommen die Agenten über den Ausgang der Auktion?
- Wie können wir plausible Testdaten generieren?

0.2.3 mögliche Fragestellungen

- Unter welchen Voraussetzungen konvergiert die Zuteilung wie?
- Was verändert die Einführung von Reverse Auctions bei Baufirmen über einzelne Skillkapazitäten als lokale Optimierung?
- Gibt es eine global optimale Lösung und wie sieht diese strukturell aus?

```
In [1]: %matplotlib inline
```

```
In [101]: import matplotlib.pyplot as plt
import numpy as np
import random
```

0.2.4 Agent/ Firma

```
In [135]: class A:
    count = 0
    def __init__(self, rs, x=(random.uniform(0,100)), y=(random.uniform(0,100))):
        self.x = x
        self.y = y
        self.rs = rs # The ammount of resources each agent has
        self._id = A.count # Its identifier
        A.count = A.count + 1

    # Format the output to the form: "A0 {R1(2), R2(4), R3(11)}"
    def __str__(self):
        return "A"+str(self._id)+" : {"+", " ".join("R"+str(i)+"("+str(x)+")" for i, x in self.rs.items())+"}"
```

```
In [136]: a = A([1,2,3])
```

```
In [137]: print(a)
```

```
A0: {R0(1), R1(2), R2(3)}
```

0.2.5 Baustelle

```
In [138]: class B:
    count = 0
    def __init__(self, rs, reward, x=(random.uniform(0,100)), y=(random.uniform(0,100))):
        self.x = x
        self.y = y
        self.rs = rs;
        self.reward = reward;
        self._id = B.count;
        B.count = B.count + 1;

    def __str__(self):
        return "B"+str(self._id)+" {"+", " ".join("R"+str(i)+"("+str(x)+")" for i, x in self.rs.items())+"}"
```

```
In [139]: b = B([3,2,1], 12)
```

```
In [151]: print(b)
```

```
B0 {R0(3), R1(2), R2(1)} +12
```

$distance : Agenten \times Baustellen \rightarrow \mathbb{R}$

$distance(a,b) := \sqrt{(a.x - b.x)^2 + (a.y - b.y)^2}$

```
In [153]: def distance(a, b):  
          return ((a.x-b.x)**2)+((a.y-b.y)**2)**0.5
```

```
In [154]: print(distance(a, b))
```

```
50.66467816514432
```