

Mechanismus Design für Austausch-Netzwerke

Jörn David, Florian Feller, Sebastian Haug, Thilo Weghorn

Ludwig-Maximilians-Universität München
Mathematisches Institut

Abstract

Dieser Essay beschreibt ein spieltheoretisches Modell für Austauschnetzwerke mit speziellem Fokus auf Filesharing-Systeme. Hierbei wurde mit Hilfe der Methoden des Mechanism Design ein Modell geschaffen, welches alle Benutzer fair behandelt. Des weiteren ist es so gestaltet, dass rational handelnde Benutzer ihren Nutzen nur dadurch maximieren können, dass sie wahre Angaben bezüglich ihrem Dateiangebot und ihrer maximalen Upload-Bandbreite angeben, was wiederum maximalen Nutzen für das gesamte System impliziert. Schließlich wurde dieses Modell mit Hilfe einer Computersimulation untersucht, wobei bestätigt wurde, dass essentielle Forderungen wie Aufrechterhaltung des Uploadstroms und Gesamtnutzenmaximierung unter diesen Modellvoraussetzungen gewährleistet werden.

1 Motivation

In der Betrachtung von asymmetrischen informationsaustauschenden Systemen mit mehreren Agenten stößt man recht schnell auf eine zentrale Problematik: sobald das Übermitteln von Informationen Kosten verursacht, führt das rationale Handeln des einzelnen zu einer Minimierung des Gesamtnutzens und somit zu einem nicht pareto-optimalen Nashgleichgewicht, in dem keine Informationen übermittelt werden. Denn: hat ein Agent keine Garantie, dass er, wenn er Informationen übermittelt, auch welche bekommt, so wird er es nicht tun. Dass sich dies nicht nur mit rationalen Agenten so verhält, sondern sich auch in der Realität keine Pareto-Verbesserung einstellt untersuchten u.a. Adar und Huberman [1].

2 Einführung in Mechanism Design

Im Gegensatz zur (klassischen) Spieltheorie, in der versucht wird, in einer gegebenen Umgebung die beste Strategie und Gleichgewichte zu finden, geht das Mechanism(us)

Design den umgekehrten Weg: Die Handlungsumgebung wird so entworfen, dass das gewünschte Verhalten ein Gleichgewicht darstellt - oder kurz: Was muss ich machen, damit sich die Spieler/Agenten so verhalten wie ich es will?

Die Teilnehmer $N = \{1, \dots, n\}$ eines Mechanismus werden Agenten genannt. Sie haben einen **Typ** $t_i \in T, i \in N$, der ihre privaten Informationen und Eigenschaften ausdrückt und eine **Strategie**, die sich aus ihrem Typ und u.U. den Strategien der anderen Agenten ergibt:

$$s_i : T \times \Sigma^{n-1} \rightarrow \Sigma, i \in N,$$

wobei Σ der Strategieraum ist. In vielen Fällen, so auch in unserem, genügt es sich auf $\Sigma = T$ und $s_i : T \rightarrow \Sigma, i \in N$ zu beschränken (mehr dazu siehe Modellierung). Mechanismen bei denen $T = \Sigma$ gilt, heißen **direkt-aufdeckende** (direct revealing) **Mechanismen** und es bietet sich die intuitive Schreibweise $\hat{t}_i := s_i(t_i)$ an.

Zunächst obliegt es dem Mechanismusbetreiber festzulegen, was sein Mechanismus erreichen soll. Um dies zu präzisieren, verwendet man eine sogenannte **Social Choice Function**:

$$f : T^n \rightarrow O$$

Hierbei ist O der Ergebnisraum, der alle möglichen Ergebnisse des Mechanismus beinhaltet. Sie wird so definiert, dass sie auf Grund der vorhandenen Typen der Agenten ein für den Mechanismusbetreiber optimales Ergebnis liefert.

Ein Mechanismus erstellt also ein Spiel samt Regeln, wozu es Eingriffsmöglichkeiten von Seiten des Mechanismus bedarf, um das Verhalten der Agenten zu beeinflussen. Im Wesentlichen stehen hier drei folgenden Methoden zur Verfügung. Dies geschieht zum einen durch Festlegung der Handlungsmöglichkeiten der Agenten, also des **Strategie-raums** Σ . Als weitere Möglichkeit kann der Mechanismus belohnend oder strafend eingreifen - dies geschieht mit Hilfe von **Ausgleichszahlungen**, die den Nutzen der jeweiligen Agenten modifizieren:

$$p_i : \Sigma^n \rightarrow \mathbb{R}, i \in N$$

Als dritte und letzte Eingriffsmöglichkeit definiert der Mechanismusbetreiber eine Ergebnisfunktion:

$$g : \Sigma^n \rightarrow O.$$

Im Gegensatz zur (ideale Ergebnisse liefernden) Social Choice Function, wird hier von den Strategien der Agenten ausgegangen, da die Typen dem Mechanismus nicht bekannt sind.

Definition: Sei $N = \{1, \dots, n\}$ eine Menge von rational handelnden Agenten mit Typen $t_1, \dots, t_n \in T$. Sei O ein Ergebnisraum. Ein Mechanismus ist ein Tupel

$$M = (\Sigma, g(\cdot), p_1(\cdot), \dots, p_n(\cdot))$$

mit Strategieraum Σ , Ergebnisfunktion $g : \Sigma^n \rightarrow O$ und Ausgleichszahlungen $p_i : \Sigma^n \rightarrow \mathbb{R}$ für alle Agenten $i \in \{1, \dots, n\}$.

Es gibt verschiedene Möglichkeiten, wie die Ausgleichszahlungen auf die **Nutzenfunktionen** der Agenten einwirken können. In unserem Kontext ist das sonst auch sehr verbreitete **quasi-lineare Nutzenmodell** relevant (mit $u_i : O \times T \times \mathbb{R} \rightarrow \mathbb{R}$)

$$u_i(o, t_i, p_i(\sigma)) = v_i(o, t_i) + p_i(\sigma),$$

wobei $v_i(o, t_i), o \in O, t_i \in T$ die unmodifizierte Nutzenfunktion von Agent i ist und $\sigma \in \Sigma^n$.

Ein Mechanismus arbeitet optimal, wenn $\forall t \in T^n : (s_1, \dots, s_n)$ ist Nashgleichgewicht $\wedge g(s_1, \dots, s_n) = f(t_1, \dots, t_n)$. In diesem Falle implementiert der Mechanismus die Social Choice Function.

3 Modellierung von Austauschnetzwerken

3.1 Austauschnetzwerke

Unter einem **Austauschnetzwerk** verstehen wir eine Menge von Individuen, die über ein Medium miteinander interagieren. Die Interaktion ist dabei von der Gestalt, dass ein Gut von einem Individuum an ein anderes weitergegeben wird. Insbesondere beschäftigen wir uns mit dem Fall, in dem die Weitergabe einer Vermehrung des Gutes entspricht (Kopie), also das abgebende Individuum das Gut nicht verliert, sondern lediglich das empfangende Individuum gewinnt. Dies ist typischerweise in Austauschnetzwerken in denen Informationen gehandelt werden der Fall. Hierbei ist jedoch zu bemerken, dass eine solche Übermittlung eines Gutes trotzdem Kosten für den Vermittler verursachen kann.

Wir konzentrieren uns hier exemplarisch auf eine spezielle Form von Austauschnetzwerken, das sogenannte **Filesharing**, da eine Modellierung nur auf einen spezifischen Fall hin maßgeschneidert sinnvoll durchgeführt werden kann.

3.1.1 Filesharing

Hierbei handelt es sich um ein Netzwerk, das den Austausch digitaler Information ("Dateien") zwischen den partizipierenden Agenten erlaubt. Hierbei wird dateibezogen und nicht agentenbezogen vorgegangen, d.h. eine nähere Kenntnis des Tauschpartners ist meist weder vorhanden noch gewünscht. Im Normalfall ist das zum Austausch verwendete Medium das Internet und die Menge der teilnehmenden Agenten nicht beschränkt - es kann theoretisch jeder Mensch, der über einen Computer und Internetzugang verfügt am Netzwerk partizipieren. Dies geschieht konkret dadurch, dass jedes Individuum durch ein Programm (ein sog. "Client") Kontakt zum Netzwerk aufnimmt. Als bekannte existierende Beispiele einiger solcher Netzwerke seien BitTorrent, eDonkey2000 oder Gnutella genannt.

3.1.2 Weitere Typen von Austauschnetzwerke

Das Konzept Austauschnetzwerk lässt sich vielfältig auf andere Situationen anwenden, so zum Beispiel auf Abteilungen eines Unternehmens, Arbeits- und Forschungsgruppen oder in einem gegenläufigen Fall das Abschreiben von Hausaufgaben unter Schülern und Studenten. Können gleichwertige Voraussetzungen und Ziele identifiziert werden, so soll unser Modell hier auch ein funktionierendes Austauschverhalten ermöglichen.

3.2 Modell

3.2.1 Aufbau

- *Zentrale Instanz*

Als Grundlage des Systems fungiert eine zentrale Instanz I , die die Austauschvorgänge überwacht und kontrollieren kann.

Es gibt Filesharing-Protokolle, die auf den ersten Blick ohne eine zentrale Instanz auszukommen scheinen, indem sie Information und Aufgaben auf die Teilnehmer verteilen. Die Überwachung und Kontrolle über das Netzwerk wird jedoch nur verteilt, nicht abgeschafft. Zudem erzeugt der komplett serverlose Betrieb ein enorme zusätzliche Netzwerkbelastung (overhead), so dass alle weit verbreiteten Filesharing Implementierungen nicht auf spezielle Serverknoten verzichten. Wir gehen der Einfachheit halber von nur einer zentralen Instanz aus, was in der Realität zwar im Allgemeinen nicht immer zu trifft, aber auch keine wirkliche Einschränkung des Modells darstellt.

- *Agenten*

Diese zentrale Instanz verwaltet die Menge N von Agenten, die am Austauschnetzwerk (hier: Filesharing

System) teilnehmen wollen.

$$N = \{1, \dots, n\}$$

Zu beachten ist, dass die zentrale Instanz hierbei weder Daten besitzt noch einfordert. Sie dient allein dazu den Austausch und die Kommunikation unter den Agenten zu regeln.

- *Zeit*

Die Zeit wird in der das System läuft wird als diskret und der Einfachheit halber endlich betrachtet:

$$Z = \{0, 1, \dots, k\}$$

Dies ist aufgrund von Rechnerarchitektur und Lebensdauer von Software im generellen gesehen realistisch. Die Gesamtlaufzeit k des Systems ist den Agenten nicht bekannt, sie ist jedoch so groß, dass ein Agent davon ausgehen kann, dass er sie in seiner Strategiewahl nicht berücksichtigen muss. Eine Modellierung mit diskreter, aber unendlicher Zeit $Z = \mathbb{Z}^+$ wäre ebenso möglich.

- *Daten*

Zu einem Zeitpunkt $\tau \in Z$ existiert im Netzwerk eine Gesamtmenge D^τ an verfügbaren Dateien, bei denen Austausch möglich ist.

Eine Datei \tilde{d} heißt zum Zeitpunkt τ verfügbar, wenn sie von mindestens einem Agenten $i \in N$ mit einer deklarierten Uploadbandbreite $\hat{b}_i^\tau > 0$ als zum Upload verfügbar deklariert wird, d.h. $\tilde{d} \in \hat{d}_i^\tau$ [siehe 4.3, Strategie].

Deklariert ein Agent $\hat{b}_i^\tau = 0$, so wird angenommen, dass für die Menge seiner zum Upload verfügbar deklarierten Dateien $\hat{d}_i^\tau = \emptyset$ gilt.

$$D^\tau = \bigcup_{i \in N} \hat{d}_i^\tau$$

Annahmen:

- Die Daten sind optimalerweise auf alle Agenten annähernd gleichverteilt. Diese Annahme ist nicht notwendig - eine ungünstige Verteilung von Daten kann jedoch dazu führen, dass das im System bald keine Tauschvorgänge mehr stattfinden.
- Die Menge der Daten zu jedem Zeitpunkt $\tau \in Z$ und damit auch generell ist endlich, was der Realität entspricht. Kein Benutzer eines Filesharing-Systems hat unendlich viel Speicher für Daten verfügbar.

- Die auszutauschenden Dateien werden als beliebig klein und von identischer Größe betrachtet. Der Einfachheit halber normieren wir diese Größe auf 1. Dies bedeutet auch, dass wir die Übertragungszeit einer Datei vernachlässigen. Dies ist keine starke Einschränkung, da große Dateien von allen Filesharing Systemen in viele kleine Teile aufgespalten werden, und die Übertragung kleiner Dateien bei heute üblichen Verbindungsgeschwindigkeiten binnen Sekunden abgewickelt ist. Diese Annahme ist nicht notwendig, nur vereinfachend. Nimmt man nicht gleiche Größe an, so sind die Kosten für einen Download nicht mehr auf 1 normiert. Nimmt man längere Übertragungszeit an, so genehmigt die Zentralinstanz Downloads während noch andere Downloads laufen. Dies ist insofern nicht problematisch, da sich hier nur temporäre Veränderungen der Bandbreite einzelner Agenten und der Verfügbarkeit einzelner Dateien ergeben, was nur mehr Aufwand im Scheduling der Downloads bedeutet.

- Die für den Upload verfügbare Bandbreite einzelner Agenten ist von derselben Größenordnung, was für die Fairness des Mechanismus in Abschnitt 4.4.1 von Bedeutung ist.

- *Anfragen*

Bei der zentralen Instanz gehen der Reihe nach Anfragen der Agenten ein. Diese werden von der Zentralinstanz nach einem beliebigen fairen Scheduling-Algorithmus $S : Z \rightarrow N$ behandelt.

Die Datei aus der derzeitigen Gesamt-Datenmenge des Systems D^τ , die ein Agent $S(\tau) \in N$, dessen Anfrage zum Zeitpunkt $\tau \in Z$ behandelt wird, von der zentralen Instanz und damit indirekt vom gesamten Netzwerk erbittet, sei mit $\tilde{d}_{S(\tau)}^\tau$ bezeichnet. Die Anfrage selbst sei ebenso bezeichnet.

- Es soll möglich sein, dass Agenten beliebig oft Anfragen an die zentrale Instanz stellen können.
- Hierbei weiß der zum Zeitpunkt $\tau \in Z$ anfragende Agent durch die zentrale Instanz von der kompletten im Netzwerk derzeit vorhandenen Datenmenge D^τ .
Hinweis: Anfragezeitpunkt und Downloadgenehmigung müssen nicht in der selben Zeitscheibe passieren, d.h. eine angefragte Datei kann u.U. bei Genehmigung schon nicht mehr verfügbar sein.

Damit ergibt sich über die komplette Zeitlinie Z eine

Anfragenmenge:

$$A = \bigcup_{\tau \in Z} \{\tilde{d}_{S(\tau)}^\tau\}$$

- *Credit-System* Um Fairness im System zu gewährleisten (siehe unten), und somit ein Ausnutzen des Systems zu unterbinden, wird die Genehmigung von Dateitransfers mit einem Credit-System geregelt. Für jeden Upload-Vorgang bekommt der Agent Credits gutgeschrieben, für jeden Download verliert er Credits. Für jeden Agenten $i \in N$ gibt es zu jedem Zeitpunkt $\tau \in Z$ die Größen c_i^τ , den aktuellen Kontostand, cr_i^τ , die Summe aller bisher erhaltenen (received) Credits und cs_i^τ , die Summe aller bisher benutzten (spent) Credits des Agenten. Zu Beginn erhält jeder Agent i $c_i = c_{start} = 1 + \delta > 1$ Credit, um einen ersten Download zu ermöglichen (hierbei gelte $1 > \delta \gg \epsilon > 0$ [siehe 4.4.1]).

In der Zentralinstanz werden cr_i^τ und cs_i^τ gespeichert, woraus sich der Kontostand zusammen mit der Kompensation in der Credit-Berechnung [siehe 4.4.1] zusammensetzt.

Insgesamt ergibt sich als Credit-System:

$$C_\tau = ((cr_1, cs_1), \dots, (cr_n, cs_n)) \in (\mathbb{R}^2)^n$$

Das Credit-System C wird nach jeder Anfrage, die genehmigt wurde, neu berechnet, bei nicht genehmigten Anfragen ändert sich jedoch nichts [siehe hierzu 4.4].

- *Zielsetzungen der teilnehmenden Agenten*

Ein Agent i der an einem Filesharingsystem teilnimmt, möchte in einem beliebigen gewählten Zeitrahmen eine Menge d^* von Dateien downloaden. Sein Nutzen ist für ihn am größten, wenn er dies mit der für ihn geringsten Aufwendung erreicht. Das Übermitteln von Dateien verursacht einem Agenten Kosten. Kosten ist hier als Antipode zu Nutzen zu verstehen. Kosten beim Filesharing sind beispielsweise die Belegung von Bandbreite oder die drohenden rechtlichen Konsequenzen bei Verletzung von Urheberrechten. Das Ziel eines rationalen Agenten ist es, die durch Upload erlangten Credits cr_i so klein wie möglich, aber so groß wie nötig zu halten. Für den **äußeren Nutzen** von Agent i zu einem Zeitpunkt $\tau \in Z$, an dem seine Dateiwünsche vorerst erfüllt sind, ergibt sich folgender Wert:

$$\hat{u}_i^\tau = cs_i^\tau - cr_i^\tau$$

wobei cs_i die vom Agenten zur Erfüllung seiner Downloadwünsche verbrauchten Credits sind. Das heißt, der Agent möchte $\hat{u}_i^\tau \geq 0$ erreichen. **Fairness des Systems** ist genau dann gegeben, wenn der äußere Nutzen gleich 0 ist, also $\hat{u}_i^\tau = 0$.

- *Entscheidungsfunktion*

Wenn nun zum Zeitpunkt $t \in Z$ die Anfrage eines Agenten $i \in N$ mit der Bitte um genau eine Datei $\tilde{d}_{S(\tau)}^\tau$ aus dem derzeitigen Dateiangebot D^τ von der zentralen Instanz bearbeitet wird, so trifft die Zentralinstanz nun mittels einer wahrheitswertigen Funktion

$$F : A \rightarrow \{ja, nein\}$$

die Entscheidung, ob der Download genehmigt wird oder nicht.

Um die Gesamtcreditzahl im System konstant zu halten [siehe 4.4.1] und Fairness zu gewährleisten, muss $\hat{u}_i \leq 0$ gelten, da ein Agent unter dieser Bedingung das System nicht ausnutzen kann. Betrachten wir hierzu die Funktion

$$F(d_{S_\tau}^\tau) = \begin{cases} ja, & \text{wenn } c_{S_\tau}^\tau \geq 1 \\ nein, & \text{wenn } c_{S_\tau}^\tau < 1 \end{cases}$$

Bei dieser Funktion sieht man zunächst, dass sie für jeden Agenten $\hat{u}_i = 1 + \delta$, $\delta > 0$ garantiert, da ja jeder Agent mehr als einen Credit Startguthaben besitzt. Nehmen wir nun realistischer Weise an, dass zukünftige Downloadmöglichkeiten für Agenten einen gewissen Wert darstellen, der mehr als einem Credit entspricht. Wir schließen Agenten i vom Download aus, die für die Zeitspanne von mindestens τ^* stets einen Kontostand in Höhe von $c_i < 1$ hatten (Die Zeitspanne τ^* ist so zu wählen, dass jeder Agent mit einer Wahrscheinlichkeit von $> 1 - \epsilon$ (mit ϵ realistisch klein) Uploads im Gesamtwert von 1 prinzipiell tätigen kann). Dies führt dann dazu, dass Agenten ihren Kontostand auf $c_{start} > 1$ halten wollen, woraus sofort $\hat{u}_i \leq 0$ folgt.

3.2.2 Dynamik des Modells

Im Folgenden beschreiben wir den dynamischen Ablauf des Filesharing-Systems. Hierbei wurde eine Darstellung als Spielbaum als weniger praktikabel angesehen als ein sequentieller Algorithmus, welcher auch unmittelbar zur Implementierung der Simulation in Abschnitt 5 führt.

1. Im Zeitpunkt $0 \in Z$ werden die zentrale Instanz sowie die Agenten initialisiert. Hierbei werden im Credit-System C der zentralen Instanz die Kontostände aller Agenten auf 1 gesetzt.
2. In den folgenden Zeitpunkten $\tau \in Z \setminus \{0\}$ können die Agenten Anfragen an das System stellen. Zu jedem Zeitpunkt kennt die zentrale Instanz die deklarierte Uploadbandbreite und die angebotene Dateimenge jedes Agenten. Stellt ein Agent eine Anfrage, so

entscheidet die zentrale Instanz anhand der Entscheidungsfunktion und des Kontostands des anfragenden Agenten, ob die Anfrage genehmigt wird.

3. Wird die Anfrage abgelehnt, wird mit Punkt 2. fortgefahren.
4. Die zentrale Instanz verteilt die Uploadlast auf die Agenten, die die jeweilige Datei anbieten, unter Berücksichtigung deren Upload-Bandbreite, wodurch der Download von mehreren Uploadern ermöglicht wird. In diesem Schritt greift nun der Mechanismus, der in Kapitel 4 ausführlich beschrieben wird.
5. Die zentrale Instanz aktualisiert ihr Credit-System wie in Kapitel 4.4.1 beschrieben. Falls das Ende der Zeitlinie Z noch nicht erreicht ist, wird mit Punkt 2. fortgefahren, andernfalls wird das Filesharing-System beendet.

Der komplette Ablauf kann in Abbildung 2 anhand eines Kontrollfluss-Diagramms, das an die Implementierung der Simulation angelehnt wurde, nachvollzogen werden.

4 Mechanismus einer Tauschanfrage

Sobald die Entscheidungsfunktion F den Download zu einem bestimmten Zeitpunkt $\tau \in Z$ freigegeben hat, wird der Mechanismus zur Abwicklung der Tauschanfrage ausgeführt. Daher wird in diesem Kapitel der Zeitpunkt der Anfrage ausgeblendet, um unnötige Indizes zu vermeiden. Möchte man den Mechanismus in den obigen Ablauf einfügen, so ergänze man alle unten genannten Variablen mit dem jeweils passenden Zeitindex.

4.1 Ziel des Mechanismus

Das Ziel des Mechanismus ist es, Agenten, die vom Netzwerk Dateien beziehen wollen, dazu zu bringen, selbst ihren Upload und vorhandene Dateien wahrheitsgemäß anzugeben, um so die Effizienz des Systems zu maximieren. Dabei sollen der Nutzen und die Kosten der Weitergabe möglichst fair über alle Teilnehmer verteilt werden. Es wird nun ein Mechanismus entworfen, der genau einen Downloadvorgang regelt.

4.2 Wertvorstellungen der Agenten

Inwiefern kann der Mechanismus die subjektiven Wertvorstellungen der Agenten antizipieren?

Prinzipiell gar nicht, da es sich um rein individuelle, teils auch in der nicht-rationellen Psyche, zumindest aber in einer dem Mechanismus nicht bekannten Information begründete Wertvorstellungen handelt. Ein Agent möchte gewisse Daten downloaden, andere eventuell nicht.

Das einzige, das der Mechanismus annehmen kann, ist, dass ein Agent prinzipiell Filesharing nutzen will - sonst wäre er nicht am System beteiligt. Daher wird der Nutzen über Downloadmöglichkeiten geregelt und der Mechanismus nimmt an, dass mehr Downloadmöglichkeit auch einen Nutzengewinn für den Agenten bedeutet.

Sollte das nicht gelten, so gibt es 2 Fälle:

- Es wird angenommen, dass ein Agent der gerade nichts downloaden will und der mit seinem Kontostand zufrieden ist, nicht am System teilnimmt, bis sich dieser Zustand wieder ändert.
- Ein Agent, der downloaden will, aber einen Kontostand hat der hoch genug ist, wird bei rationalem Handeln seinen Upload sperren.

Ein Eingreifen hier ist in diesen Fällen durch den Mechanismus nicht möglich, da er dem Agenten keine Anreize bieten kann und aufgrund mangelnder Informationen über tatsächlich verfügbare Bandbreite und Dateimenge auch nicht strafen kann. Außerhalb des Mechanismus können hier jedoch Maßnahmen ergriffen werden um die Häufigkeit dieses Falls zu verringern (z.B. Verfall des Downloadguthabens durch "Abzinsung" [siehe 4.4]).

4.3 Modellierung

In diesem Abschnitt wird der Mechanismus

$$M = (\Sigma, g(\cdot), p_1(\cdot), \dots, p_n(\cdot))$$

betrachtet, der einen einzelnen, von der Zentralinstanz genehmigten Download der Datei $\tilde{d} \in D$ (D repräsentiert die Menge der Dateien) optimiert, indem die Anzahl der Uploader und deren Bandbreite maximiert werden. Wir definieren nun die dazu erforderlichen Konzepte:

- Typ
 - Ein Agent $i \in N = \{1, \dots, n\}$ hat in jeder Zeiteinheit einen Typ $t_i = (d_i, b_i) \in T = \mathbb{P}(D) \times \mathbb{R}^+$, wobei D die Menge der filesharingrelevanten Dateien eines Agenten ist und die für den Upload verfügbare Bandbreite als reelle Zahl modelliert wird.
- Strategie
 - Die Strategie ist im betrachteten Mechanismus *direct revealing*, d.h. $\Sigma = T, \hat{t}_i : T \times \Sigma^{n-1} \rightarrow \Sigma$. Ein Agent gibt seinen Typ $\hat{t}_i = (\hat{d}_i, \hat{b}_i)$ an.
 - Ein Agent kann jedoch auch einen Typ angeben, der weniger Dateien oder eine geringere Bandbreite ausweist. Es wird angenommen, dass technische Überprüfungsmaßnahmen existieren, die es verhindern, dass ein Agent eine größere Bandbreite oder mehr Dateien angibt als dies tatsächlich der Fall ist.

- Ergebnis(raum)

Der Ergebnisraum gibt Auskunft darüber, von welchen Agenten die Datei heruntergeladen wird.

- Der Ergebnisraum O hat folgende Gestalt:

$$O = \{o \in [0, 1]^n : (\forall i \in \{1, \dots, n\}) : I_{d_i}(\tilde{d}) = 0 \Rightarrow o_i = 0) \wedge \sum_{i=1}^n o_i = 1\}$$

Hierbei ist $o = (o_1, \dots, o_n)$ und I_{d_j} die Indikatorfunktion, die angibt, ob Agent j die gewünschte Datei hat oder nicht.

- Interpretation: jedem vorhandenen/angegebenen Typ eines jeweiligen Agenten wird eine Zahl zwischen 0 und 1 zugeordnet. Hat der Typ die Datei nicht, so ist dies Null. Die Summe all dieser Zahlen ist 1. Der Ergebnisraum beschreibt, wie viel Prozent der Datei von welchem Agenten heruntergeladen werden.

- Social Choice Function

- Die Social Choice Function wählt $o \in O$ so, dass die Gesamtbandbreite maximal wird (und die Übertragungszeit somit minimal).
- Wir wählen für $f : T^n \rightarrow O$:

$$f(t) = \left(\frac{I_{d_i}(\tilde{d}) \cdot b_i}{\sum_{j=1}^n I_{d_j}(\tilde{d}) \cdot b_j} \right)_{i \in \{1, \dots, n\}}$$

- Interpretation: Jeder Agent der die Datei hat bekommt seinen prozentualen Anteil an der verfügbaren Gesamtbandbreite zugewiesen. Ist die Bandbreite des Downloaders größer als die Summe der Bandbreiten der Uploader, so wird die Bandbreite aller Uploader hierdurch voll ausgenutzt. Ist dies nicht so, so ist die Belastung der Uploader prozentual identisch.
- Alternativ ließe sich die Verteilung auch absolut anstatt prozentual konstruieren - hierbei wäre die Auslastung der Bandbreiten (z.B. in kb/s) gleich, solange die individuelle Auslastung unter der maximalen individuellen Uploadbandbreite läge. Ist ein Agent voll ausgelastet, wird die Restlast unter den verbleibenden, nicht voll ausgelasteten Agenten bis zum Erreichen der Downloadbandbreite verteilt. Unter der Annahme einer unbegrenzten Downloadbandbreite sind diese Ansätze offensichtlich äquivalent. Die Entscheidung fiel hier auf das prozentuale Modell, da so auch bei geringer Downloadbandbreite der

Anreiz die eigene Bandbreite zu erhöhen gegeben ist, falls der Mechanismus dafür sorgt, dass Uploaden einen positiven Nutzen für den Agenten hat.

- Ergebnisfunktion

- Sie soll der Social Choice Function so nahe wie möglich kommen, hat jedoch im Gegensatz zur SCF nur Zugriff auf die Strategien, nicht auf die Typen.
- D.h. für $g : \Sigma^n \rightarrow O$ gilt:

$$g(\hat{t}) = \left(\frac{I_{d_i}(\tilde{d}) \cdot \hat{b}_i}{\sum_{j=1}^n I_{d_j}(\tilde{d}) \cdot \hat{b}_j} \right)_{i \in \{1, \dots, n\}}$$

- Nutzenfunktion und Ausgleichszahlungen

- Upload per se bringt keinen Nutzen, d.h. für die unmodifizierte Nutzenfunktion $v_i : O \rightarrow \mathbb{R}$ gilt

$$v_i(g(\hat{t})) \leq 0$$

Wir beschränken uns hier bewusst auf eine Abhängigkeit von den Strategien der Agenten und vernachlässigen die Abhängigkeit von den Typen, da wir es hier mit einem direkt-enthüllenden Mechanismus zu tun haben und wir annehmen, dass ein linearer Zusammenhang zwischen Uploadleistung und Kosten besteht, d.h. dass $|v_i|$ auf Grund der Abhängigkeit von $g(\hat{t})$ mit wachsendem \hat{b}_i steigt und b_i , außer als Maximum der \hat{b}_i , hier nicht mit eingeht.

- Offensichtlich können sich die Ausgleichszahlungen aus Sicht des Mechanismus, der den wahren Typ der Agenten nicht kennt, nur nach den Strategien richten, also $p_i : \Sigma^n \rightarrow \mathbb{R}$.
- Damit teilgenommen wird, muss für die individuelle Nutzenfunktion $u_i : O \times \mathbb{R} \rightarrow \mathbb{R}$ nun gelten, dass

$$u_i(g(\hat{t}), p_i(\hat{t})) \geq 0$$

- Hieraus folgt bei quasilinearer Nutzenfunktion $u_i(g(\hat{t}), p_i(\hat{t})) = v_i(g(\hat{t})) + p_i(\hat{t})$ direkt für das individuelle Payment p_i , dass

$$p_i(\hat{t}) \geq -v_i(g(\hat{t}))$$

- Implementation der Social Choice Function

- Wie bereits in 4.2 beschrieben, kann der Mechanismus Agenten, deren Kontostand für ihr empfinden hoch genug ist, nicht dazu bringen, ihre

wahre Bandbreite oder Dateimenge zu offenbaren, da er ihnen nichts anzubieten hat und eine Abstrafung aufgrund mangelnder Information über ihren wahren Typ nicht möglich ist. Also ist die Social Choice Function hier nicht implementierbar.

- Betrachten wir also den Fall, dass am Mechanismus nur Agenten teilnehmen, die auch ein höheres Downloadguthaben erreichen wollen.

Wie angenommen, bedingt ein Anwachsen der verwendeten Upload-Bandbreite ein Anwachsen der individuellen Kosten v_i (und umgekehrt). Auf Grund der Wahl der Social Choice Function f und der Entscheidungsfunktion g , impliziert eine größere Angabe an verfügbarer Bandbreite auch eine größere Beteiligung an der Uploadleistung. Daher bedingt ein Anwachsen der angegebenen freien Upload-Bandbreite \hat{b}_i ein Anwachsen der individuellen Kosten v_i (und umgekehrt). Gilt $p_i = -v_i$, so ist der Gesamtnutzen eines Agenten immer 0. Dies gibt dem Agenten keinen Grund seine Bandbreite falsch anzugeben - es gibt ihm jedoch auch keinen Grund sie wahrheitsgemäß anzugeben, da sein Nutzen konstant ist. Der Nutzen des Agenten soll aber abhängig von der mitgeteilten Bandbreite größer werden, d.h. $v_i = 0 \Rightarrow u_i = 0$ und $v_i < 0 \Rightarrow u_i > -v_i$.

Bei dem hier verwendeten linearen Nutzenmodell entspricht das einer multiplikativen Konstante $c = 1 + \epsilon > 1$ vor $-v_i$ als Payment, d.h.

$$u_i = I_{\hat{d}_i}(\tilde{d}) \cdot (v_i + c \cdot (-v_i)).$$

Denn ist dies der Fall, so wächst u_i mit wachsendem $|v_i|$, denn für $\tilde{v}_i < v_i < 0$, $\tilde{u}_i = I_{\hat{d}_i}(\tilde{d}) \cdot (\tilde{v}_i + c \cdot (-\tilde{v}_i))$ und $I_{\hat{d}_i}(\tilde{d}) = 1$ gilt: $\tilde{u}_i - u_i = \tilde{v}_i + c \cdot (-\tilde{v}_i) - v_i - c \cdot (-v_i) = (1-c) \cdot (\tilde{v}_i - v_i) = -\epsilon \cdot (\tilde{v}_i - v_i) = \epsilon \cdot (v_i - \tilde{v}_i) > 0$.

Behauptung: Ist $c > 1$, so ist $\forall i \in N : \hat{t}_i = t_i$ Nash-Gleichgewicht.

Beweis: Da mit $c > 1 \Rightarrow \epsilon > 0$ die Nutzenfunktion mit zunehmenden $|v_i|$ wächst und $|v_i|$ wiederum mit wachsendem \hat{b}_i steigt (siehe [Nutzenfunktion und Ausgleichszahlungen]), wächst die Nutzenfunktion mit wachsendem \hat{b}_i . Dies bedeutet nun, mit $\hat{t}^{*b_i} = \hat{t} \in \Sigma^n$ bis auf $\pi_i(\hat{t}^{*b_i}) = (\hat{d}_i, b_i)$, dass

$$\begin{aligned} \max_{\hat{b}_i \in \pi_b(\Sigma) = \mathbb{R}^+} u_i(g(\hat{t}_i), p_i(\hat{t}_i)) &= \\ u_i(g(\hat{t}_i^{*b_i}), p_i(\hat{t}_i^{*b_i})) & \end{aligned}$$

Da die Teilnahme an einem Upload davon abhängt, ob man die angefragte Datei besitzt und

auch angibt sie zu besitzen, gilt mit $\hat{t}^{*d_i} = \hat{t} \in \Sigma^n$ bis auf $\pi_i(\hat{t}^{*d_i}) = (d_i, \hat{b}_i)$, dass

$$u_i(g(\hat{t}_i), p_i(\hat{t}_i)) \leq u_i(g(\hat{t}_i^{*d_i}), p_i(\hat{t}_i^{*d_i})).$$

Hierbei herrscht Gleichheit, wenn \tilde{d} gleichzeitig sowohl in d_i als auch in \hat{d}_i vorhanden oder nicht vorhanden ist.

Nun folgt direkt, mit $\hat{t}^* = \hat{t} \in \Sigma^n$ bis auf $\pi_i(\hat{t}^*) = (d_i, b_i) = t_i$, dass

$$\max_{\hat{t}_i \in \Sigma} (u_i(g(\hat{t}_i), p_i(\hat{t}_i))) = u_i(g(\hat{t}^*), p_i(\hat{t}^*)).$$

Hieraus folgt die Behauptung. \square

Bemerkung: Solange $\tilde{d} \in \hat{d}_i$ ist, lassen sich bzgl. der Dateideklaration auch Falschaussagen tätigen ohne die Optimalität bezüglich dieser Tauschanfrage zu beeinflussen. Da jedoch angenommen wird, dass nur relevante Dateien zur Verfügung gestellt werden, führt eine solche Angabe dazu, dass andere Tauschanfragen nicht optimal für den Agenten genutzt werden. Also führt die wiederholte Ausführung des Mechanismus bzgl. unterschiedlicher Dateien dazu, dass - da wir uns auf relevante Dateien eingeschränkt haben - eine komplett wahrheitsgemäße Angabe der verfügbaren Dateien optimale Strategie im Sinne des "übergeordneten" Systems ist.

Folgerung: Da ein rationaler Agent also $\hat{t}_i = t_i$ wählt folgt aus der Definition der Entscheidungsfunktion und der der Social Choice Function direkt, dass $f(t) = g(\hat{t})$ gilt, wobei $T^n \ni t = \hat{t} \in \Sigma^n$. Mit obiger Behauptung folgt somit, dass der Mechanismus in diesem Fall die Social Choice Function implementiert.

4.4 Einbettung in das Gesamtmodell

4.4.1 Berechnung des aktuellen Creditstands

Wie im Beweis der Implementation der Social Choice Function gesehen, muss, damit Anreizkompatibilität gegeben ist, das Payment die Form $p_i = (1 + \epsilon) \cdot (-v_i)$ haben. Dies bedeutet für jeden Agenten i , dass $c_i^{\tau+1} = c_i^\tau + (1 + \epsilon) \cdot (-v_i)$. Hieraus folgt direkt, dass die Summe über die Credits aller Agenten $C = \sum_{i=1}^n c_i$ nach einem Mechanismsdurchlauf um $\epsilon \cdot |\tilde{d}|$ gewachsen ist, wobei $|\tilde{d}|$ die Dateigröße von \tilde{d} ist.

Behauptung: Wächst oder fällt C unbeschränkt so bricht das System zusammen (d.h. der Austausch kommt unwiderruflich zum Erliegen bzw. erreicht eine unakzeptabel niedrige Austauschrate).

Beweis: Steigt C unbeschränkt, so bricht das System früher oder später zusammen, denn irgendwann hat jeder Agent den Stand an Credits erreicht, den er benötigt und er wird somit seinen Upload sperren. Fällt C unbeschränkt, so

werden aufgrund der Entscheidungsfunktion immer weniger Downloads genehmigt und die Austauschrate nimmt ab, bis sie im Grenzübergang zu unendlicher Zeit verschwindet, also fast sicher kein Austausch mehr stattfindet. \square

Folgerung: Es muss eine obere und untere Schranke für C geben, damit der Betrieb aufrecht erhalten werden kann. Der Einfachheit und Transparenz halber wählen wir für beide genau den Wert $n \cdot c_{start}$ wobei n die Anzahl der am System in beteiligten Agenten und c_{start} der Startwert an Credits eines jeden Agenten ist.

Eine Möglichkeit dies zu gewährleisten ist, nach jedem Mechanismsdurchlauf eine Absenkung der Credits aller Agenten vorzunehmen:

$$\forall i \in N : c_i^{\tau+1} = c_i^\tau - \frac{\epsilon}{n}$$

Wobei $|\tilde{d}|$ die Dateigröße von \tilde{d} nach der in den Annahmen vorausgesetzten Normierung 1 ist. Dies liefert, wie man leicht sieht, das Gewünschte.

Bemerkung: Durch die Absenkung folgt, dass ein Agent der am Upload teilnimmt und einen Upload-Anteil zugewiesen bekommt der kleiner als $\frac{1}{n}$ ist, insgesamt trotzdem noch Verlust macht. Dies wurde jedoch durch die Modellannahmen in Abschnitt 3.2.1 ausgeschlossen.

Folgerung: Hiermit ergibt sich nun für einen Agenten i zum Zeitpunkt $\tau^* \in Z$ folgende Formel für die bisher durch Upload erhaltenen Credits:

$$cr_i^{\tau^*} = \sum_{\tau=1}^{\tau^*} \epsilon \cdot (-v_i^\tau)$$

und folgende Formel für den aktuellen Kontostand:

$$c_i^{\tau^*} = \underbrace{(c_{start} + cr_i^{\tau^*} - cs_i^{\tau^*})}_{\geq 0} - |M^{\tau^*}| \cdot \frac{\epsilon}{n}$$

wobei $|M^{\tau^*}|$ die Anzahl der Mechanismsdurchläufe im Zeitraum $t = 0, 1, \dots, \tau^*$ ist, also die Anzahl der bisher erfolgten Downloads.

4.4.2 Wirkung

Durch die Implementation der Social Choice Funktion wird ein Agent i bis zum Erlangen der individuell benötigten Menge Credits wahrheitsgemäß seinen Typ deklarieren, d.h. solange der Mechanismus eine Handhabe hat, sorgt er für austauschförderliches Verhalten. Hierdurch werden im Rahmen des Möglichen die beiden Ziele der **maximalen Dateivielzahl und Übertragungsgeschwindigkeit** gewährleistet, ohne das dritte, durch die Entscheidungsfunktion erfüllte Designkriterium **Fairness** maßgeblich zu beeinflussen.

5 Simulation

5.1 Darstellung des Aufbaus

Das EVOSIMULATION Simulations-Framework besteht im Wesentlichen aus den folgenden Objekten (Klassen):

- **Administration**
Dieses Kontrollobjekt ist das Herz der Simulation, das den unter Abschnitt 4 beschriebenen Mechanismus ausführt. Hier werden die Populationen (Population) mit ihren Individuen (Individual) erzeugt und die Simulation in Form einer gestartet (vgl. Zustandsdiagramm).
- **Individual**
Dieses Objekt stellt einen Teilnehmer des Austauschnetzwerks dar, der mit einer vorgegebenen Strategie (siehe Type) am Mechanismus teilnimmt.
- **Population**
Dies repräsentiert die Gesamtheit aller Individuen, die jeweils einen unterschiedlichen Typ besitzen können (siehe Type).
- **Type**
Der Typ eines Individuums bestimmt dessen Upload-Strategie. Derzeit implementiert das Framework die Typen Cooperative, Defective und Rationale. Für die Dauer der Simulation, ändert ein Individuum seinen Typ nicht, denn dieser ist fest kodiert. Nichtsdestotrotz kann eine Population in Gruppen von Individuen unterschiedlichen Typs partitioniert sein.
 - **Cooperative**
Der kooperative Typ teilt seine Dateien bedingungslos mit den anderen Agenten und beantwortet somit jede Downloadanfrage positiv – mit der maximal verfügbaren Uploadrate.
 - **Defective**
Der defektierende Typ beantwortet Downloadanfragen niemals positiv, sondern lädt nur seinerseits von kooperierenden (Cooperative) oder rationalen Agenten (Rationale) herunter.
 - **Rationale**
Dieses Individuum stoppt seine Uploadbereitschaft bei Erreichen einer frei wählbaren, aber festen Credit-Obergrenze (hier MaxCredits=2, erlaubt 2 Dateidownloads). Wenn durch folgende Downloads dieses Individuums die Downloadcredits wieder unter diese Grenze gesunken sind, gibt es den Upload wieder frei und erhält somit wieder Credits, usw.

- **DecisionFunction**
Das Interface DecisionFunction legt die Funktionalität der Entscheidungsfunktion so fest, wie sie in Abschnitt 3.2.1 definiert wurde. Es stehen mehrere konkrete Implementierungen zur Verfügung.
- **UtilityFunction**
Definiert den äußeren Nutzen eines Individuums über den zeitlichen Verlauf des Filesharings hinweg. Es stehen zwei konkrete Implementierungen zur Verfügung.

Die Subsystemzerlegung wird in Abbildung 1 dargestellt. Das dynamische Modell des Frameworks wird in Abbildung 2 beschrieben.

5.2 Auswertung

Um die Funktionsweise des Mechanismus zu veranschaulichen, wurden folgende Kenngrößen simuliert, die das (simulierte) Austauschnetzwerk aus externer Sicht quantitativ beschreiben:

- Durchschnittliche Dateianzahl im Zeitverlauf jeweils gemittelt über alle Individuen eines Typs: Abb. 3
- Durchschnittlicher Kontostand (Creditzahl) jeweils gemittelt über alle Individuen eines Typs: Abb. 4
- Durchschnittlicher äußerer Nutzen der verschiedenen Individentypen jeweils gemittelt: Abb. 5 und 6.

Für alle Auswertungen wurde die deterministische Entscheidungsfunction F verwendet, wie in Abschnitt 3.2.1 eingeführt. In den folgenden Abbildungen 3 bis 6 werden die Ergebnisse in Form von Zeitreihen dargestellt.

Die Simulation zeigt, dass der Mechanismus der Aufrechterhaltung des Filesharings dient.

6 Fazit und Ausblick

In diesem Artikel wurde zunächst ein abstrakter Mechanismus zur Koordination des Austauschs von Objekten über ein Netzwerk vorgestellt. Der Mechanismus sorgt dafür, dass alle download-willigen Agenten ihr Dateiangebot sowie ihre Bandbreite maximieren, womit auch der Gesamtnutzen des Filesharing-Systems maximiert wird. Dies wird durch die in Abschnitt 5 durchgeführte Simulation empirisch belegt, die den vorgestellten Mechanismus implementiert.

Schließlich bleiben noch Stabilitätsbetrachtungen durchzuführen, die zeigen sollen, wie sich das Gesamtsystem z.B. unter Variation der Datei- und Nutzerverteilung verhält. Mögliche Erweiterungen des Filesharing-Systems und dessen Mechanismus, die u.a. deren Realitätsbezug erhöhen, werden im Folgenden kurz benannt.

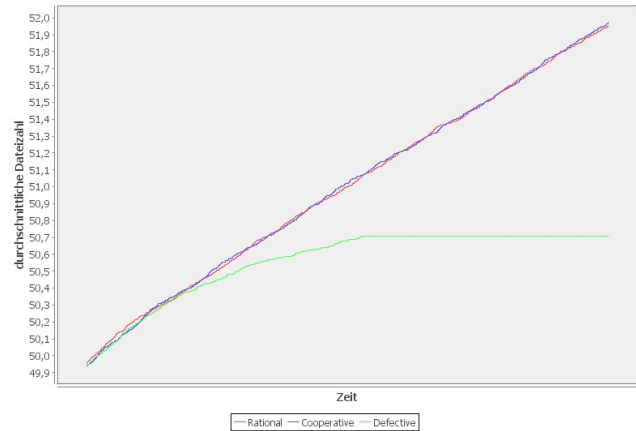


Abbildung 3. Mittlere Dateianzahl: Die Individentypen Rationale (rot) und Cooperative (blau) verfügen im Simulationsverlauf ca. über die gleiche Anzahl an Dateien. In der ersten Hälfte der Simulation dominiert der Typ Rationale den Typ Cooperative sogar leicht in punkto Dateianzahl. Deutlich erkennbar ist die frühe Stagnation der Dateianzahl für den Typ Defective (grün).

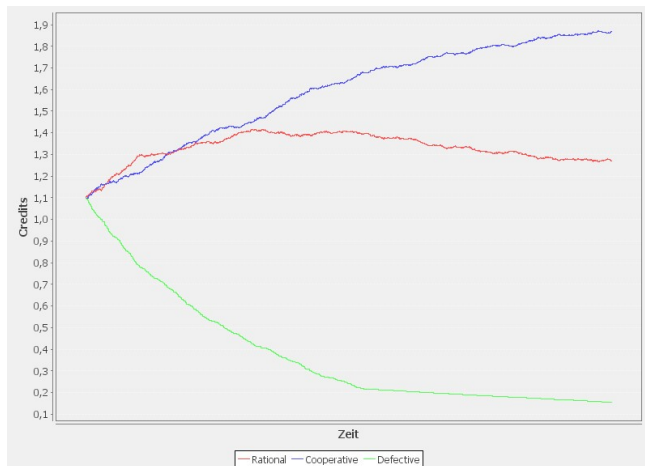


Abbildung 4. Individuen vom Typ Cooperative (blau) erzielen im Verlauf der Simulation (2000 Runden) die höchste Creditzahl c_i^T , gefolgt vom Typ Rationale (rot). Die Creditzahl des Typs Defective (grün) geht hingegen asymptotisch gegen 0.

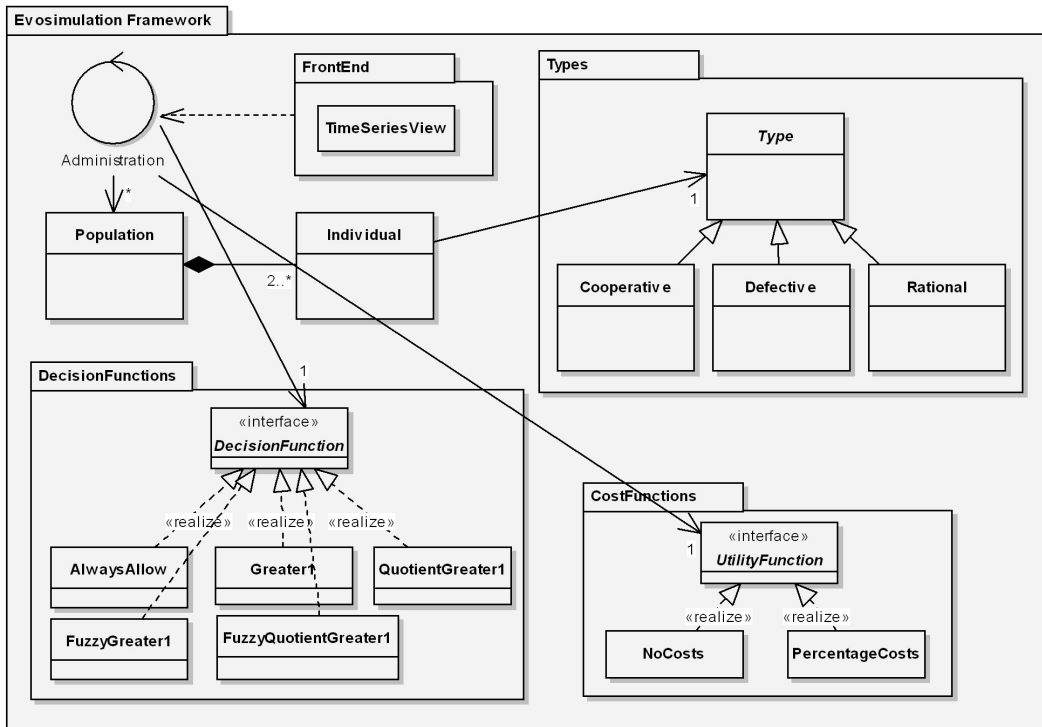


Abbildung 1. Subsystemzerlegung des Evosimulation Frameworks (UML-Kassendiagramm).

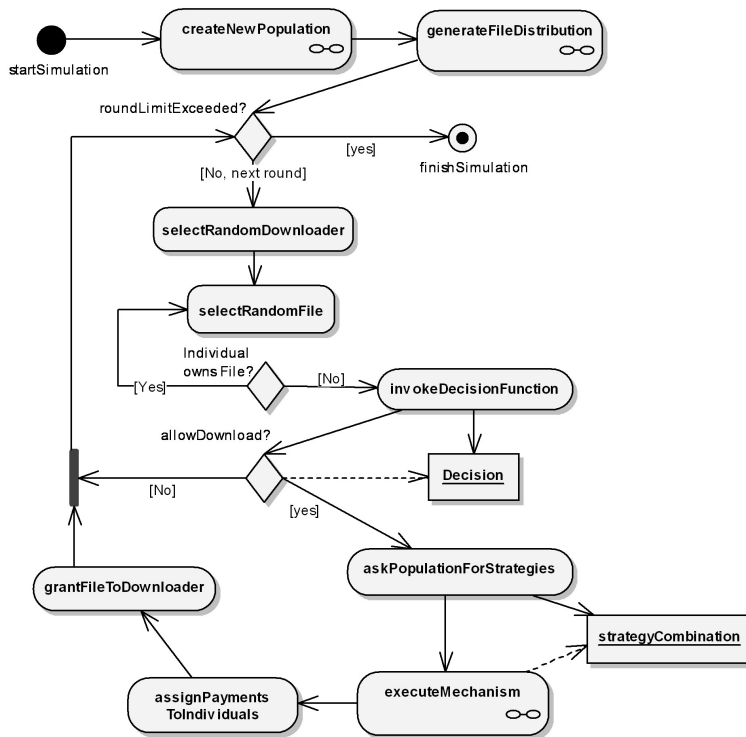


Abbildung 2. Kontrollfluss des Simulationsalgorithmus' (UML-Kontrollflussdiagramm).

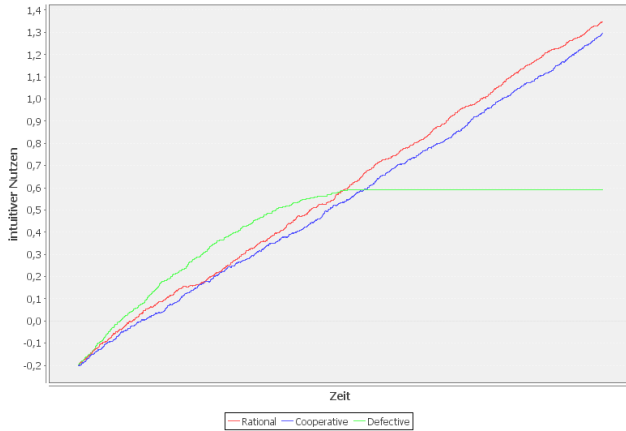


Abbildung 5. Intuitiver Nutzen $cr - f \cdot cs$ der verschiedenen Individuentypen, der vom Mechanismus über die Runden der Simulation hinweg induziert wird, d.h. unter Verwendung einer zentralen Instanz. Der Unterschied zum durchschnittlichen Kontostand in Abbildung 4 besteht darin, dass hier tatsächlich der rationale Typ (Rationale, rot) die anderen dominiert, da dieser seinen (kostenverursachenden) Upload auf 0 reduziert, sobald genügend Download-credits angehäuft sind.

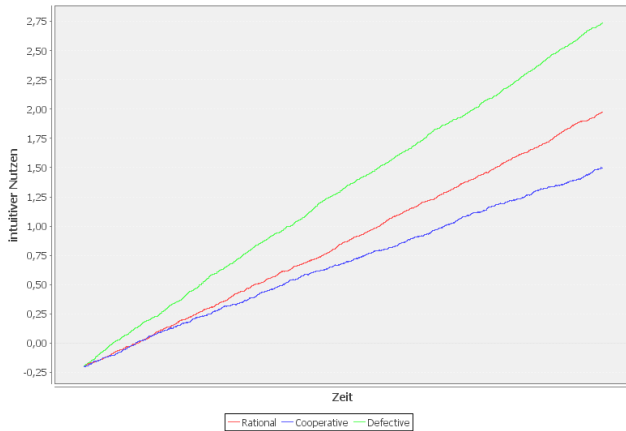


Abbildung 6. Intuitiver Nutzen $cr - f \cdot cs$ ohne Verwendung einer zentralen Instanz und somit ohne Regelung durch den Mechanismus (unter Verwendung der Entscheidungsfunktion AlwaysAllow, siehe Abb. 1). Erwartungsgemäß “gewinnt” hier der Typ Defective (grün), der niemals kooperiert (bzgl. seinem Upload). Der Typ Rationale verhält sich in diesem Kontext offensichtlich nicht rational, denn er lädt ungefähr soviel Dateivolumen hoch, wie er herunterlädt.

- Zulassen einer variablen Teilnehmermenge.
- Berücksichtigung von Dateiübertragungen, die sich auf mehrere Runden erstrecken.
- Untersuchung von verschiedenen Angebots- und Nachfrageverteilungen bzgl. der auszutauschenden Objekte. Damit soll auch die Homogenitätsannahme bzgl. der auszutauschenden Objekte fallen gelassen werden.

Der letztgenannte Punkt ist dabei von hoher Praxisrelevanz, da man bei der Nachfrage (evtl. auch beim Angebot) – abweichend von der bisherigen Simulation – eher von einer Normalverteilung der Dateien ausgehen muss. Es können sich beispielsweise “Flaschenhälse” im Dateiaustausch ergeben, wenn selten im Netzwerk vorkommende Dateien häufig nachgefragt werden.

Literatur

- [1] Eytan Adar and Bernardo Huberman, *Free riding on gnutella*, First Monday **5** (2000), no. 10.
- [2] Vincent Conitzer and Tuomas Sandholm, *Incremental mechanism design*, Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.
- [3] Rajdeep K. Dash, Nicholas R. Jennings, and David C. Parkes, *Computational- mechanism design: A call to arms*, IEEE INTELLIGENT SYSTEMS, IEEE Computer Society, 2003.
- [4] Jason D. Hartline and Tim Roughgarden, *Optimal mechanism design and money burning*, Proceedings of the 40th annual ACM symposium on Theory of computing, Victoria, British Columbia, Canada, ACM New York, NY, USA, 2008, pp. 75–84.
- [5] Kevin Lai, Michal Feldman, Ion Stoica, and John Chuang, *Incentives for cooperation in peer-to-peer networks*, Workshop on Economics of Peer-to-Peer Systems, 2003.
- [6] Jürgen Steimle, *Algorithmic Mechanism Design - Eine Einführung*, Informatik im Fokus, Springer Berlin, 2008.