

Einleitung

Eine der wesentlichen Neuerungen, die das Internet in die Informatik einführt, besteht in seiner sozioökonomischen Komplexität. Es handelt sich beim Internet nicht um ein homogenes Netzwerk, das von einer einzigen Organisation betrieben oder gesteuert wird, sondern um ein komplexes Zusammenspiel verschiedener Teilnetze, die Besitzern mit unterschiedlichen ökonomischen Interessen gehören. Das Internet hat somit nicht nur die Charakteristika eines Computernetzes, sondern auch die eines Wirtschaftssystems.

Aufgrund dieser sozioökonomischen Komplexität stellen sich grundlegend neue Herausforderungen im Entwurfsprozess der Informatik. Traditionelle Rechnernetze wurden üblicherweise von einer einzigen Organisation, wie einer Forschungseinrichtung oder einem Unternehmen, betrieben. Es konnte daher davon ausgegangen werden, dass Algorithmen und Protokolle von allen – mit Ausnahme von fehlerhaften — Teilsystemen so befolgt werden, wie es vom Entwickler vorgesehen ist. Im Internet mit seiner Vielzahl von ökonomisch eigenständigen Teilnetzen muss man dagegen annehmen, dass die Akteure sich so verhalten, wie

es für ihre jeweiligen Besitzer am vorteilhaftesten ist – sie handeln eigennützig und damit nicht unbedingt kooperativ.

Zugleich ergeben sich durch den hohen Grad an Vernetzung zunehmend mehr Szenarien, in denen im Internet eine kooperative Problemlösung nötig ist. Das Routing von Nachrichten ist ein klassisches Beispiel. In zukünftigen Anwendungen könnten schnelle Rechner freie Rechenkapazitäten für CPU-intensive Berechnungen zur Verfügung stellen und große Datenmengen könnten automatisch auf Rechnern mit freiem Festplattenplatz gespeichert werden.

Der Mangel an Kooperationsbereitschaft, der dem eigennützigen Verhalten der Akteure entspringt, kann dazu führen, dass die Akteure einen Algorithmus oder ein Protokoll nicht so befolgen, wie es vom Entwickler vorgesehen ist. Dies führt häufig zu Ergebnissen, die aus Sicht des Systemdesigners und des Gesamtsystems ineffizient sind. So zeigt beispielsweise eine Untersuchung, dass sich in der Peer-to-Peer-Filesharing-Börse Gnutella ca. 70 % der Anwender als Free-Rider verhalten [2], d.h. zwar von den Angeboten der anderen Teilnehmer profitieren, selbst jedoch keine Dateien bereitstellen. Aus Sicht des Systemdesigners, in die wir uns im Folgenden begeben, ist Eigennützigkeit ein Hindernis, das überwunden werden soll, um bessere Ergebnisse zu erzielen. Der zu entwerfende Algorithmus oder das zu entwerfende Protokoll soll robust gegenüber eigennützigen Manipulationsversuchen der Agenten sein.

Algorithmic Mechanism Design

Algorithmic Mechanism Design (AMD) ist ein Untergebiet von Algorithmik, Spieltheorie und Mikroökonomie. Sein Einsatzfeld sind Situationen, in denen nicht-kooperative Akteure gemeinsame Problemlösungen treffen müssen. Dabei untersucht es die Frage, wie garantiert werden kann,

dass sich die eigennützigen Akteure an ein vorgesehenes Protokoll halten. AMD nützt dafür den Egoismus der Agenten aus und schafft Anreize für ein korrektes Verhalten: Ein eigennütziger Akteur würde ein insgesamt effizientes Ergebnis zunächst nicht unterstützen, wenn es in seinen Auswirkungen für ihn individuell nachteilig wäre. Wenn nun aber ein Mechanismus eingerichtet wird, der in einem solchen Fall eine Art von Belohnungsleistung an den Akteur vornimmt, z. B. in Form einer Geldzahlung, die seinen individuellen Nachteil bei diesem Ergebnis ausgleicht, so wird der Akteur das insgesamt effiziente Ergebnis ebenfalls bevorzugen und sich somit kooperativ verhalten. Somit werden die Interessen des Systemdesigners und die jedes einzelnen Akteurs auf eine Linie gebracht – es ist dann für die eigennützigen Akteure von Vorteil, sich genau so zu verhalten, wie es der Systemdesigner vorgesehen hat.

Klassisches Mechanismusdesign, das aus der Ökonomie stammt, beschäftigte sich nicht mit Fragestellungen der Informatik. Wichtig war ausschließlich die Überlegung, wie in einem Gemeinwesen effiziente Entscheidungen getroffen werden können. Die algorithmischen Eigenschaften der Mechanismen war dagegen unerheblich. Klassisches Mechanismusdesign wird in der Ökonomie bereits seit über 40 Jahren untersucht. (Kürzlich wurde sogar der Wirtschaftsnobelpreis an die Hauptvertreter der Forschung zum Mechanismusdesign verliehen.) Eingang in die Informatik fand dieses Gebiet als Algorithmic Mechanism Design aber erst gegen Ende des vergangenen Jahrzehnts, als nicht-kooperative Netzwerke zunehmend an Bedeutung gewannen. Dieses Forschungsfeld untersucht die für die Informatik zentralen Fragen der Komplexität von Mechanismen und wie ein Mechanismus effizient algorithmisch implementiert werden kann. Inzwischen wurden zahlreiche Anwendungsfälle dieses neu entstandenen Algorithmic Mechanism Design untersucht.

Dazu gehören u.a. Routing [69, 47, 31, 37, 35, 3, 38, 93, 96], Ressourcen-Sharing in Peer-to-peer-Netzwerken [44, 92, 40] oder das Design von Auktionen für digitale Güter [55, 43, 9].

Mechanismusdesign wird auch als „inverse Spieltheorie“ bezeichnet. Während die Spieltheorie Situationen mit nicht-kooperativen Multiagentensystemen betrachtet und untersucht, welche Eigenschaften die entstehenden Ergebnisse besitzen, geht Mechanismusdesign in die entgegengesetzte Richtung: Hier wird von bestimmten gewünschten Ergebniseigenschaften ausgehend untersucht, wie die „Spielregeln“ gestaltet werden müssen, damit nicht-kooperative Akteure dieses gewünschte Ergebnis realisieren. Ein Beispiel für einen sehr einfachen Mechanismus mit simplen Spielregeln wäre die Ergebnisbestimmung durch Münzwurf. Dieser einfache Ansatz garantiert aber nicht, dass das Ergebnis bestimmte gewünschte Eigenschaften besitzt. Auktionen sind ein anderes Beispiel für Mechanismen. Diesen liegen komplexere „Spielregeln“ zugrunde, die dazu führen, dass das Ergebnis einer Versteigerung eher den Wünschen der Teilnehmer entspricht als die Zufallsentscheidung beim Münzwurf.

Ein Beispielszenario

Betrachten wir die Problemstellung des Routing im Internet. Damit ein Datenpaket vom Sender zum Empfänger gelangt, muss es häufig durch viele Teilnetze, die Dritten gehören, weitergeleitet werden. Da bisher vergleichsweise geringe Datenmengen anfielen, erfolgte die Weiterleitung ohne die Berechnung von Kosten. Durch die Zunahme des Datenverkehrs für bandbreitenintensive Multimedia-Kommunikation und damit verbundene Quality-of-Service-Garantien vermuten Experten, dass diese Praxis nicht beibehalten wird. Stattdessen könnte sich zunehmend ein wirt-

schaftliches Kalkül durchsetzen, bei dem für die Weiterleitung von Datenpaketen bezahlt werden muss – die Teilnetze verhalten sich nicht mehr kooperativ, sondern eigennützig. Wäre das Internet von einer einzigen Organisation betrieben, würde für die Weiterleitung eines Datenpakets derjenige Weg gewählt, bei dem die geringsten *tatsächlichen* Kosten entstehen. Die eigennützigen Teilnetze werden jedoch taktische Preise für die Weiterleitung festlegen, die nicht unbedingt den ihnen dabei entstehenden wahren Kosten entsprechen. Der Sender, der den Weg wählt, der für *ihn* am kostengünstigsten ist, läuft damit Gefahr, sein Paket über einen ineffizienten Weg zu schicken, der der *Gemeinschaft* der Teilnetze höhere Kosten verursacht als nötig. Die Einrichtung eines Mechanismus kann dieses Problem beheben, indem der Mechanismus in den „Spielregeln“ bestimmte Ausgleichszahlungen so definiert, dass die Teilnetze einen Anreiz haben, ihre wahren Kosten zu deklarieren.

Zum Aufbau dieses Buches

Dieses Buch hat zum Ziel, den Leser in dieses Forschungsgebiet von wachsender Bedeutung einzuführen, wobei sowohl grundlegende Fragen als auch Lösungsansätze für zentrale Probleme, die sich im AMD ergeben, dargestellt werden sollen. Dafür wurden die aktuellen Forschungspublikationen – hauptsächlich Veröffentlichungen der Tagungsbände der Association for Computing Machinery (ACM) – in Zusammenhang gebracht, grundlegende Ergebnisse dargestellt und, soweit möglich, anhand von Beispielen von Internetanwendungen veranschaulicht. Zu bemerken bleibt, dass diese Einführung nicht den Anspruch hat, sämtliche in der Forschung untersuchten Gebiete von AMD vorzustellen. So werden u.a. der Ansatz, Computertechniken zum Finden von guten Mechanismen einzusetzen (Automated

Mechanism Design) und der Entwurf von Mechanismen für dynamische Multiagentensysteme nicht diskutiert.

Das folgende Kapitel führt anhand des Beispiels von Auktionen und von Shortest-Path-Routing zunächst in die Grundlagen des Algorithmic Mechanism Design ein. Dabei werden sowohl die ökonomischen und spieltheoretischen Konzepte als auch die für die theoretische Informatik zentralen Komplexitätsüberlegungen dargestellt.

Im dritten Kapitel werden wir sehen, dass es wichtige Mechanismen gibt, die nicht effizient algorithmisch implementierbar sind. Am Beispiel der kombinatorischen Auktion soll dargestellt werden, welche theoretischen Probleme auftreten und welche konzeptionellen Lösungen denkbar sind.

Distributed Mechanism Design, eine neue Forschungsrichtung im Algorithmic Mechanism Design, untersucht die Gestaltung von verteilten Mechanismen. Dieser Ansatz, der für viele Anwendungen in dezentralen Netzwerken wie dem Internet besonders geeignet erscheint, soll schließlich im letzten Kapitel vorgestellt werden. Durch die verteilte Berechnung ergeben sich – speziell in Peer-to-Peer-Netzwerken – neue Problematiken, für die aktuelle Lösungsansätze aufgezeigt werden sollen. In diesem letzten Kapitel wird das Routing-Beispiel des zweiten Kapitels wieder aufgegriffen. Diesmal steht jedoch die Nähe zur Praxis im Vordergrund. Ausgehend vom Border-Gateway-Protokoll, das im Internet derzeit das Standardprotokoll für Interdomain Routing ist, soll ein verteilter Mechanismus vorgestellt werden, der im Wesentlichen zu dem ihm zugrunde liegenden Border-Gateway-Protokoll kompatibel ist.



<http://www.springer.com/978-3-540-76401-4>

Algorithmic Mechanism Design

Eine Einführung

Steimle, J.

2008, X, 167 S. 28 Abb., Softcover

ISBN: 978-3-540-76401-4