Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARTIN GRÖTSCHEL SVEN O. KRUMKE JÖRG RAMBAU

Wo bleibt der Aufzug?

Preprint SC 99-29 (August 1999)

WO BLEIBT DER AUFZUG?

MARTIN GRÖTSCHEL, SVEN O. KRUMKE UND JÖRG RAMBAU

1. Einleitung

Ein Problem, dem wir täglich begegnen: Morgens verschlafen, unterwegs alle Ampeln rot, und dann kommt im Bürogebäude der Aufzug nicht. Welcher Idiot hat wohl diese Steuerung entworfen? Das muß man doch leicht mit OR lösen können!

Wirklich?

Daß dies nicht so einfach ist, wie man zunächst denkt, soll dieser Artikel verdeutlichen. Wir erläutern anhand des Aufzugbeispiels die wesentlichen Fragestellungen und Analysemethoden bei Online- und Echtzeit-Optimierung. Dieser Artikel ist kein breiter Übersichtsartikel. Wir wollen hier nur ein sich gerade entwickelndes Forschungsgebiet vorstellen. Die Steuerung von Aufzügen ist gut geeignet, um die Bedeutung der Online- und Echtzeit-Optimierung aufzuzeigen und offene Forschungsfragen zu erklären.

Versetzen wir uns in die Lage des Aufzugs, der gerade im achten Stock steht. Soeben hat A im siebten Stock einen Transport nach unten angefordert, B im dritten Stock nach oben und C im vierundzwanzigsten nach unten. Wir haben nur sehr wenig Zeit zu überlegen („Echtzeit"), damit der Geduldsfaden der Wartenden nicht reißt. Zum Glück sind wir mit Mitteln des OR in der Lage, für die drei anliegenden Aufträge einen „bestmöglichen" Fahrplan schnell zu erstellen. Doch als wir gerade in der achten Etage losgefahren sind, drückt D auf der achten Etage auf den Knopf ... Was nun? Zurückfahren? Was, wenn nach Aufsammeln von D in der achten Etage auch noch E, F, G, ... mitfahren wollen? Aber wann holen wir dann C im Penthouse ab? Also sollen wir doch lieber D warten lassen, bis alle bisherigen Aufträge abgearbeitet sind und D sauer wird, weil wir schon zum wiederholten Mal an seine Etage ungnädig vorbeirauschen? Wir sind gerade auf die sogenannte Online-Eigenschaft des Aufzugproblems gestoßen.

Traditioneller OR-Philosophie folgend beruht unser Anfangs-Fahrplan darauf, daß wir einen kürzesten Transportplan für die drei bekannten Fahraufträge bestimmen. Unser Problem besteht darin, daß wir zukünftige Fahraufträge nicht kennen und selbst ein Reoptimieren bei jedem neuen Fahrauftrag insgesamt nicht zu einem global gesehen kürzesten Fahrplan führen muß.

2. Online- und Echtzeit-Optimierung

In der „klassischen" kombinatorischen Optimierung geht man davon aus, daß die Daten jeder Probleminstanz vollständig gegeben sind. Aufbauend auf diesem vollständigen Wissen berechnet dann ein Algorithmus eine optimale oder approximative Lösung. In vielen Fällen modelliert diese *Offline Optimierung* jedoch die Anwendungssituationen nur ungenügend.

Zahlreiche Problemstellungen in der Praxis sind in natürlicher Weise *online*: Sie erfordern Entscheidungen, die unmittelbar und ohne Wissen zukünftiger Ereignisse getroffen werden müssen. Beispiele für natürliche Online-Probleme sind das Routing und Verteilen von Telefongesprächen, Fuhrparkmanagement, die Steuerung von Aufzügen oder das Paging in virtuellen Speichersystemen.

Bei der Online-Optimierung modelliert man die Eingabedaten als Anfrage-Sequenz $\sigma = (r_1, \ldots, r_m)$. Ein Online-Algorithmus muß eine Anfrage r_i bei ihrem Auftreten bearbeiten (oder zur Bearbeitung verplanen), bevor er die nächste Anfrage r_{i+1} erhält. Im allgemeinen fordert man, daß die vom Online-Algorithmus getroffenen Entscheidungen nicht widerrufen werden können (Varianten gibt es natürlich, sie sind auch in der Literatur zu finden).

In diesem Sinne ist das Aufzugsteuerungsproblem ein Online-Problem mit Echtzeitanforderungen. Der Aufzug muß nach einer beschränkten Rechenzeit auf neue Anforderungen reagieren; einmal getroffene Entscheidungen können nicht revidiert werden, auch wenn sie sich in der Zukunft als ungünstig herausstellen.

Während die Online-Problematik nur von der jeweiligen Aufgabenstellung abhängt, sind Echtzeit-Anforderungen zusätzliche Nebenbedingungen, die unter anderem von den aktuell verfügbaren Rechnerresourcen abhängen.

3. Zielfunktionen

Kehren wir zu unserem Aufzugproblem zurück. Um überhaupt Lösungen sinnvoll bewerten zu können, stellt sich zunächst die Frage nach einer geeigneten Zielfunktion. Bei Personenaufzügen hätte man gerne zufriedene Benutzer. Schon hier muß man sich die Frage stellen, ob man die individuelle Zufriedenheit der Fahrgäste dem Allgemeinwohl opfern möchte: es ist leicht, Beispiele zu finden, in denen der global kürzeste Fahrplan eine Person sehr lange warten läßt. Der „gerechteste Algorithmus", der die Fahrgäste nach dem First-Come-First-Serve-Prinzip bedient, liefert jedoch bei stärkerer Auslastung so ineffiziente Fahrpläne, daß auch die individuellen Wartezeiten untragbar werden.

4. Analysekonzepte

Lange Zeit war offen, wie man die Qualität eines Online-Algorithmus sinnvoll bewertet. Themen wie Aufzugsteuerung und Steuerung von Call-Centern werden in der Warteschlangentheorie behandelt. Diese beschäftigt sich aber vor allem mit der Dimensionierung von Kapazitäten. Bei bestimmten Annahmen über die Verteilung von Aufträgen und bei Existenz eines festgelegten Algorithmus können Aussagen über die erwartete Aufenthaltszeit in einer Warteschlange, die durchschnittliche Länge einer Warteschlange u. ä. gemacht werden. Warteschlangentheorie ist dann adäquat, wenn der Abarbeitungsalgorithmus kanonisch gegeben ist, z. B. First-Come-First-Serve.

Zur Beurteilung von Online-Algorithmen hat sich den letzten Jahren *kompetitive Analyse* (siehe [BEY98, FW98]) als ein Standardmittel durchgesetzt. Dabei vergleicht man den Zielfunktionswert $C_A(\sigma)$ einer vom Online-Algorithmus A

generierten Lösung bei Eingabe einer Anfrage-Sequenz σ mit dem Wert $C_{\mathrm{OPT}}(\sigma)$ einer optimalen Offline-Lösung. Der Algorithmus A heißt dann c-kompetitiv, wenn es eine Konstante b gibt, so daß für alle Anfrage-Sequenzen σ gilt: $C_A(\sigma) \leq c \cdot C_{\mathrm{OPT}}(\sigma) + b$. Interessante Ergebnisse sind in diesem Zusammenhang

- der Entwurf von Algorithmen mit „kleinem" Kompetitivitätsfaktor c und
- untere Schranken für c, d. h. finde $c > 1$, so daß kein c-kompetitiver Algorithmus für das Problem existiert.

Man beachte, daß hierbei den Algorithmen keinerlei Beschränkungen der Zeit- oder Speicherkomplexität auferlegt werden. Die „Schwierigkeit" eines Online-Problems beruht ausschließlich auf der unvollständigen Informationslage.

Für die Optimierung der Gesamtbearbeitungsszeit („Makespan") bei einem Aufzug mit Kapazität 1 konnte in [AKR98] ein 5/2-kompetitiver Algorithmus gefunden werden. Zugleich konnte gezeigt werden, daß kein deterministischer Algorithmus besser als $1 + \sqrt{2}/2 \approx 1.70$ kompetitiv sein kann. Betrachten wir weitere interessante Zielfunktionen (z. B. durchschnittliche/maximale Wartezeit) sieht man jedoch schnell ein, daß kein Algorithmus kompetitiv sein kann: selbst im „trivialen Fall" von nur einem Aufzug mit Kapazität eins können für jedes noch so große c Beispiele konstruiert werden, so daß jeder Online-Algorithmus mindestens die c-fache durchschnittliche/maximale Wartezeit der optimalen Offline-Lösung benötigt. Daraus folgt, daß man theoretisch nicht das garantieren kann, was man in der Praxis erwartet: zügige Abfertigung aller „Fahrstuhlkunden". Das heißt, so leicht wie gedacht, kann man das morgentliche Fahrstuhlproblem auch mit den besten OR-Methoden nicht lösen.

Für andere Online-Probleme wie Bin-Packing oder Paging existieren in der Literatur recht starke Kompetitivitätsresultate mit sehr dicht beieinanderliegenden (oder sogar identischen) oberen und unteren Schranken (siehe [FW98]). Im Zusammenhang mit Logistikproblemen finden sich jedoch kaum nichttriviale Resultate. Im Gegenteil, in [Kam97] wurde das erstaunliche Ergebnis bewiesen, daß für ein bestimmtes Order-Batching Problem (Einsammeln von Glückwunschkarten, ein reales Problem aus der Praxis) alle Algorithmen den gleichen trivialen Kompetitivitätsfaktor haben (siehe auch die Kommentare zur „Triviality Barrier" in [FW98]).

5. Simulation

Die theoretischen Resultate sind teilweise so schwach, daß sie für die Praxis kaum Entscheidungshilfen liefern. Häufig ist der einzige Ausweg Validierung und Evaluierung durch Simulation.

Grundlage für eine aussagekräftige Simulation ist Zugang zu realistischen Daten. Für verschiedene Aufzugsysteme wurde aufbauend auf AMSEL [AMS97] ein Simulationssystem entwickelt, mit dem das Verhalten der Algorithmen in praktischen Situationen evaluiert werden kann. Einige der für die Zielfunktion „Makespan" kompetitiven Algorithmen zeigten dabei auch gutes praktisches Verhalten für andere Zielfunktionen, die für die Praxis interessanter sind.

Dies ist nicht nur bei Aufzugsteuerung so. Praxisnahe Simulation ist *das* Mittel, mit dem die „Qualität" von Online-Algorithmen bestimmt wird. Erst hierbei wird entschieden, welche Algorithmen in Softwaresystemen in den praktischen Einsatz gelangen.

6. Fazit

Während kompetitive Analyse alleine oftmals zuwenig Information über die Leistungsfähigkeit von Algorithmen liefert, gewinnt man doch Einsicht in die Problemstruktur. Die Kombination aus theoretischer Analyse und Simulation führt dann häufig auch zu praktikablen Algorithmen, deren Lösungen (durch Simulation empirisch bewertet) erheblich besser sind als man theoretisch garantieren kann.

Unser Fahrstuhlproblem vom Anfang kann man in der Praxis also durchaus zufriedenstellend bewältigen, wenn man anhand repräsentativer Datensätze anwendungsspezifische Online-Heuristiken entwirft. Gegen unerwartet massiven Andrang, z.B. direkt vor Arbeitsbeginn, oder „bewußt böswillige“ Datensätze ist jedoch auch der beste Algorithmus machtlos.

Literatur

[AKR98] N. Ascheuer, S. O. Krumke, and J. Rambau, *Competitive scheduling of elevators*, Preprint SC 98-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin, November 1998.

[AMS97] *Amsel—a modelling and simulation environment library*, 1997, Entwickelt am Konrad-Zuse-Zentrum Berlin. Online-Dokumentation unter `http://www.zib.de/ascheuer/AMSEL`.

[BEY98] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, 1998.

[FW98] A. Fiat and G. J. Woeginger (eds.), *Online algorithms: The state of the art*, Lecture Notes in Computer Science, vol. 1442, Springer, 1998.

[Kam97] Nicola Kamin, *On-line optimization of order picking in an automated warehouse*, Ph.D. thesis, Tech. Univ. Berlin, 1997.

Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, 14195 Berlin-Dahlem, Germany.

E-mail address: {groetschel,krumke,rambau}@zib.de