Aspekte der Mathematik — Anwendungen
Vorlesungsskriptum WS 16/17

Christian Krattenthaler

Inhaltsverzeichnis

Einführung

In diesem Teil der Vorlesung geht es um Anwendungen der Mathematik. Da zu diesem Zeitpunkt außer den einfachsten Grundlagen der Mathematik nichts vorausgesetzt werden kann, ist es nicht möglich, Anwendungen in größerer Tiefe hier zu besprechen. Es soll bloß ein erster Eindruck von (einigen wenigen) Anwendungen der Mathematik gegeben werden.

Es ist ja (leider) so, dass viele Schülerinnen und Schüler die Schule mit dem Eindruck verlassen, dass sie nun die ganze Mathematik gelernt haben, beziehungsweise dass Mathematik etwas Abgeschlossenes ist, wo es nichts mehr Neues geben kann.[1] Schlimmer noch, dass nämlich Mathematik etwas völlig Nutzloses ist, das im täglichen Leben gar nicht benötigt wird — wenn es etwas zu rechnen gibt, dann erledigt das der Computer, und im übrigen ist Technik das Um und Auf heutzutage.

Es gibt nichts „Falscheres“ als diesen Eindruck. Das Gegenteil ist wahr. Mathematik ist überall, insbesondere in unserer heutigen technisierten Zeit. Wenn wir einen Computer benutzen, wenn wir ein Mobiltelephon benutzen, das Internet, in ein Flugzeug steigen, den Wetterbericht hören, usw., es ist immer Mathematik im Spiel. Naturwissenschaften und Technik *bauen auf und benötigen für ihre Weiterentwicklung essentiell mathematische Erkenntnisse*. So ist es auch klar, dass neue technische Entwicklungen und Erkenntnisse der Naturwissenschaften ständig neue mathematische Herausforderungen stellen, die dann Gegenstand der mathematischen Forschung sind.

Als Mittelschullehrer(in) der Mathematik muss man daher das Ziel haben, diese letztere Tatsache den Schülerinnen und Schülern zu vermitteln (und nicht die Nutzlosigkeit der Mathematik). Im Idealfall sollten außerdem die faszinierenden Seiten der Mathematik weitergegeben werden. Dass das nicht leicht ist, da Mathematik unbestreitbar nicht immer leicht verständlich und somit nicht immer leicht zu erklären ist, das ist kein Geheimnis. Nichtsdestotrotz muss am Ende der Schulausbildung das Bewusstsein stehen, dass Mathematik nicht abstrakt und langweilig, sondern im Gegenteil eine überaus kreative und faszinierende Wissenschaft ist, die überall im täglichen Leben — explizit oder implizit — zu finden ist und ihre Anwendung findet.

Werden wir konkreter: Welche Anwendungsgebiete der Mathematik gibt es, und welche mathematischen Theorien kommen dort zur Anwendung? Hier ist ein kurzer — selbstverständlich extrem unvollständiger — Überblick.

• Physikalische Prozesse, chemische Prozesse, Evolutionsprozesse werden durch *Differentialgleichungen* beschrieben. Aus diesem Grund ist die mathematische Analyse und numerische Auswertung und Simulation von Differentialgleichungen von so essentieller

[1] Standardreaktion auf das „Bekenntnis“ Mathematiker zu sein: „Ach so? Was gibt es denn da zu forschen? Ist nicht alles schon bekannt“

Bedeutung. Da uns zu diesem Zeitpunkt alle Grundlagen dazu fehlen, kann dies hier nicht näher besprochen werden.

• In vielen Zusammenhängen kommt zusätzlich eine „Zufallskomponente“ ins Spiel, insbesondere in der *Finanzmathematik*; andere Beispiele wären das Testen von Hypothesen („Ist dieses Medikament wirksam oder nicht?“; „Ist diese Umfrage signifikant oder nicht?“) Dann kommen Methoden der *Wahrscheinlichkeitsrechnung* und *Stochastik* zur Anwendung, manchmal auch in Kombination mit Differentialgleichungen (sogenannte *stochastische Differentialgleichungen*, die in der Finanzmathematik eine enorm wichtige Rolle spielen). Auch auf diese Dinge kann leider wegen mangelnder Voraussetzungen hier nicht näher eingegangen werden.

• Analyse und Rekonstruktion von Bildern oder Klang führt uns zur *Fourieranalyse*, oder allgemeiner zur *Harmonischen Analyse*. Dies ist ebenfalls wegen seiner Anwendungsrelevanz ein intensivst studiertes Gebiet. Auch hier fehlen uns alle Grundlagen, sodass wir nicht näher darauf eingehen können.

• Unternehmensentscheidungen werden auf Grund von Nebenbedingungen, die durch begrenzte Ressourcen, zeitliche und gesetzliche Rahmenbedingungen, usw. gefällt. Ziel ist es immer, eine bestimmte Größe („Gewinn“, „Kosten“, ...) — oder auch mehrere gleichzeitig — zu optimieren (maximieren/minimieren). Wir haben es also mit Problemen der *Optimierung* zu tun. Im Prinzip unterscheidet man zwischen *stetiger* und *diskreter* (*kombinatorischer*) *Optimierung*. Optimierungsentscheidungen im Unternehmensbereich können mit Hilfe von stetiger Optimierung gelöst werden.[2] Wir werden diese hier in ihrer einfachsten Form kennenlernen: die *lineare Optimierung*, siehe Kapitel 1.

• Früher hatte man einmal ausgedruckte Fahrpläne für Züge, Busse, usw. Wenn man eine kompliziertere Reise unternehmen musste, konnte es sehr mühsam sein, sich die besten Verbindungen herauszusuchen. Heute macht man Routenplanung durch Eintippen des Startortes A und des Zielortes B auf der Internetsite der ÖBB, der Wiener Verkehrsbetriebe, eines Fluganbieters, usw. Innerhalb von Sekundenbruchteilen erhält man die schnellste oder billigste Verbindung von A nach B. Dies ist ein typisches Problem der *kombinatorischen Optimierung*, nämlich das *Problem des kürzesten Weges*. Wir werden dieses in Kapitel 2. besprechen.

• Jeder von uns tippt täglich Stichworte in Suchmaschinen ein, und in Sekundenbruchteilen wird uns eine Liste von Internetsites serviert, wo dieses Stichwort vorkommt. Natürlich läuft auch hier im Hintergrund Mathematik ab, nämlich ein Suchalgorithmus (und noch ein paar andere — mathematische — Dinge). Suchalgorithmen gehören zum größeren Gebiet der *Informationstheorie*. Wir werden in Kapitel 3 das allererste fundamentale Faktum der Informationstheorie, die sogenannte *informationstheoretische Schranke*, kennenlernen.

• Jedes Mal, wenn wir das Internet benutzen, wenn wir telephonieren, wenn wir ein Computernetzwerk benutzen, dann werden Daten übertragen. Diese werden über Medien wie Kabel oder auch die Luft übertragen. Das läuft nicht hundertprozentig ab,

[2]Das gilt selbst dann, wenn wir eigentlich Lösungen in den natürlichen Zahlen suchen, wie etwa Stückzahlen. Es wird nämlich um große Zahlen gehen, und dann reicht es das stetige Optimierungsproblem zu lösen und anschließend gegebenfalls zu runden.

da werden Fehler auftreten. Wie man es trotzdem erreichen kann, dass ausgesandte Nachrichten ohne Fehler beim Empfänger ankommen, das ist ein großes Thema der *Kodierungstheorie*: wie können Übertragungsfehler korrigiert werden?

In manchen Zusammenhängen ist ein weiteres wichtiges Thema die *Datensicherheit*. Wenn wir gerade ein Passwort oder eine Kreditkartennummer eingeben, wollen wir nicht, dass jemand von außen „mithören“ kann. Wie das „Abhören“ verhindert (oder zumindest: sehr erschwert) werden kann, damit beschäftigt sich die *Kryptographie*. Auf Letzteres, nämlich die Kryptographie — und hier insbesondere auf die berühmte RSA-Verschlüsselung — wird in Kapitel 4 eingegangen werden. Für die Kodierungstheorie bräuchte man ein wenig Vorkenntnisse aus der Linearen Algebra, deshalb kann sie hier nicht besprochen werden.

KAPITEL 1

Lineare Optimierung — Das Simplex-Verfahren

Unter *Linearer Optimierung* versteht man die Lösung von Optimierungsaufgaben, wo sowohl die Funktion, die optimiert werden soll, als auch die Nebenbedingungen, unter denen sie optimiert werden soll, linear in den Variablen des problems sind.

Beginnen wir mit einem typischen Beispiel.

Beispiel. Ein Unternehmer stellt zwei Produkte, A und B, her. Ein Stück des Produktes A ist 2 Kilogramm schwer, ein Stück des Produktes B wiegt 1 Kilogramm. Pro Tag können höchstens 1200 Stück des Produktes A erzeugt werden, und auch höchstens 1200 Stück des Produktes B. Einmal pro Tag liefert der Unternehmer seine Ware an einen Markt, und zwar mit einem Transporter, der 3200 Kilogramm Kapazität hat. Er kann jedes Stück der Produkte A und B mit 100 Euro Gewinn verkaufen. Wieviel Stück der beiden Produkte A und B soll der Unternehmer auf den Transporter laden, um maximalen Gewinn zu erzielen?

Der Punkt hier ist nicht, dass es sich hierbei um ein besonders praxisnahes Problem handelt. Es soll aber illustrieren und bewusst machen, dass zahlreiche Probleme der Praxis (die viel komplexer sein werden) in den Bereich der linearen Optimierung fallen. In diesem Kapitel soll besprochen werden, wie man diesen Typ von Problemen löst.

Übersetzen wir das obige Problem zuerst einmal in mathematische Sprache. Sei x die Stückanzahl des Produktes A und y die Stückanzahl des Produktes B, die auf den Transporter geladen werden. Aus der Angabe ergeben sich die Bedingungen

$$2x + y \leq 3200 \tag{1.1}$$
$$x \leq 1200 \tag{1.2}$$
$$y \leq 1200 \tag{1.3}$$
$$x, y \geq 0. \tag{1.4}$$

Unter diesen Voraussetzungen wollen wir den Gewinn $100x + 100y$ maximieren. Das ist natürlich dazu äquivalent, die Funktion $\psi(x, y) = x + y$ zu maximieren.

Zunächst werden wir das obige Problem geometrisch lösen. Wir werden daraus die Intuition für eine allgemeine Lösungsstrategie gewinnen, die wir dann anschließend besprechen werden.

Überlegen wir, wie der Bereich der *zulässigen* Paare (x, y) — also der Bereich jener Paare (x, y), die (1.1) erfüllen — aussieht. Das lässt sich in der Ebene darstellen, siehe Figur 1. Der zulässige Bereich wird von den Geraden $2x+y = 3200$, $x = 1200$, $y = 1200$ und den Koordinatenachsen eingegrenzt. In anderen Worten: Es handelt sich um das Polygon mit den Eckpunkten $(0, 0)$, $(1200, 0)$, $(1200, 800)$, $(1000, 1200)$, $(0, 1200)$, die als Schnittpunkte der begrenzenden Geraden entstehen, und dessen Inneres.

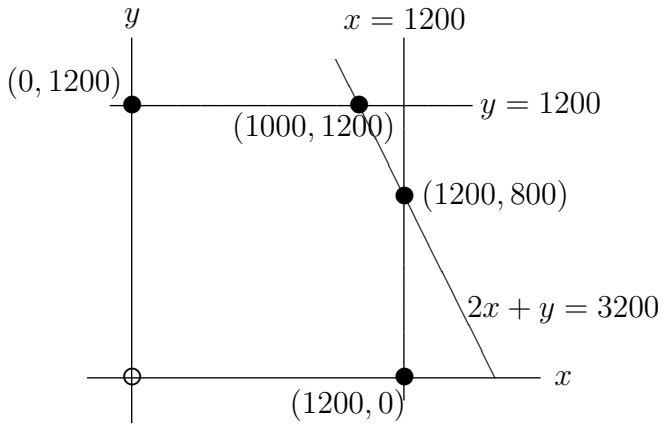

ABBILDUNG 1. Der zulässige Bereich

Nun müssen wir uns auch die *Zielfunktion* — die Funktion, die wir optimieren wollen — geometrisch vor Augen führen. Zur Erinnerung: In unserem Beispiel wollen wir die Funktion $\psi(x, y) = x + y$ maximieren. Figur 2 zeigt alle Punkte mit $x + y = c$, wo c irgendeine gegebene reelle Zahl ist.

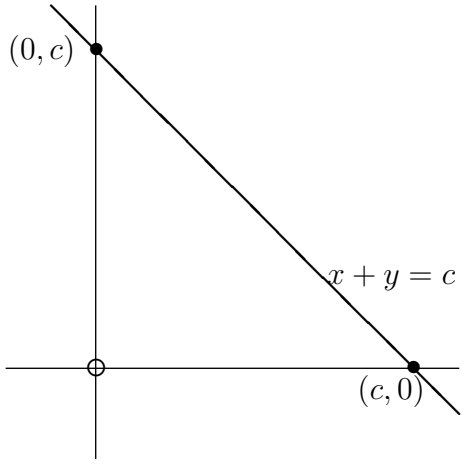

ABBILDUNG 2. Die Zielfunktion

Es ist aus dieser Figur klar, dass, je größer die Konstante c wird, die Gerade $x+y = c$ weiter nach rechts/oben parallel verschoben wird.

Darum geht es also: Wir müssen im zulässigen Bereich in Figur 1 jene(n) Punkt(e) finden, die auf einer Gerade $x + y = c$ mit c möglichst größ liegen. Diese finden wir dadurch, dass wir zunächst $c = 0$ setzen, also die Gerade $x+y = 0$ betrachten, und diese dann immer weiter nach rechts/oben parallel verschieben, so lange noch mindestens ein Punkt des zulässigen Bereiches darauf liegt. In Figur 3 sieht man, dass man die Gerade bis $x + y = 2200$ verschieben kann, sodass gerade noch der Punkt $(1000, 1200)$ daraufliegt. Der Unternehmer wird also 1000 Stück des Produktes A und 1200 Stück des Produktes B laden, mit denen er einen maximalen Gewinn von 220000 Euro machen kann.

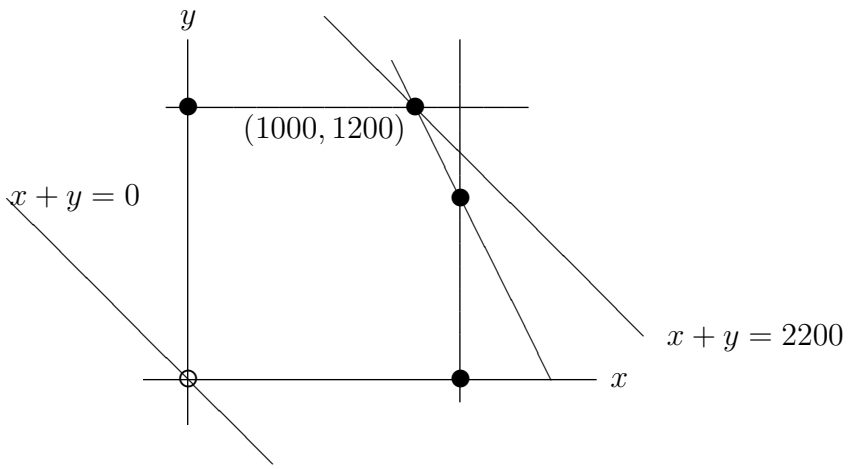

ABBILDUNG 3. Das Maximum

Wir fassen zusammen: Der zulässige Bereich ist von Geraden beschränkt; zur Ermittlung des Optimums verschieben wir die Zielfunktion so weit, so lange noch ein Punkt des zulässigen Bereiches darauf liegt. Es ist klar, dass dadurch das Optimum (so es existiert) immer in einem Eckpunkt des zulässigen Bereiches gefunden werden kann.

Motivation/Herleitung des Simplex–Verfahrens

Das Simplex–Verfahren wurde 1947 von George B. Dantzig vorgestellt. Es löst Optimierungsprobleme mit linearen Einschränkungen und linearer Zielfunktion. Solche Probleme werden deswegen Probleme der „Linearen Optimierung“ genannt.

Betrachten wir das folgende Problem: Man will das *Minimum* der Funktion

$$\psi(x) = -350x_1 - 260x_2 - 100x_3$$

finden (was äquivalent zum Finden des *Maximums* von $350x_1+260x_2+100x_3$ ist), wobei die Bedingungen

$$3x_1 + x_3 \leq 30 \tag{1.5}$$
$$3x_1 + 6x_2 + 2x_3 \leq 120 \tag{1.6}$$
$$10x_1 + 5x_2 + x_3 \leq 120 \tag{1.7}$$
$$x_1, x_2, x_3 \geq 0, \tag{1.8}$$

erfüllt sein sollen. Das System (1.5)–(1.8) definiert offensichtlich einen konvexen Polyeder,[1] und (intuitiv) wird das Minimum der Zielfunktion $\psi(x)$ (so es existiert) in einem Eckpunkt des Polyeders zu finden sein. Die Idee des Simplex–Verfahrens ist es, in einem Eckpunkt zu beginnen, und sich dann entlang von Kanten des Polyeders sukzessive zu anderen Eckpunkten zu „bewegen“, bis man einen Eckpunkt erreicht, wo die Zielfunktion das Maximum annimmt.

Der Polyeder, der durch (1.5)–(1.8) beschrieben wird, hat einen offensichtlichen Eckpunkt: den Punkt $(0, 0, 0)$. Es ist ebenfalls klar, dass man, ausgehend von $(0, 0, 0)$, drei

[1]Eine Teilmenge des $\mathbb{R}^d$, die von Hyperebenen begrenzt wird, nennt man *Polyeder*. Sollte so ein Polyeder *beschränkt* sein, dann spricht man von einem *Polytop*.

Möglichkeiten hat, sich entlang von Kanten weiterzubewegen: Man kann entlang der x_1-Achse „marschieren“, entlang der x_2-Achse, und entlang der x_3-Achse (alles in positiver Richtung). Man sieht, dass sich die Zielfunktion $\psi(x)$ in allen drei Fällen verkleinern wird (was man ja anstrebt). Die Zielfunktion wird sich am stärksten verkleinern, wenn man sich entlang der x_1-Richtung bewegt. Die Bedingungen (1.5)–(1.8) sagen uns, wieweit wir uns bewegen können: Die Bedingung (1.5) beschränkt x_1 mit $30/3 = 10$, die Bedingung (1.6) beschränkt x_1 mit $120/3 = 40$, und die Bedingung (1.7) beschränkt x_1 mit $120/10 = 12$. Wir können uns also bis $x_1 = 10$ bewegen. Wir erreichen so den Eckpunkt $(10, 0, 0)$. Die Zielfunktion ist nun gleich -3500 geworden.

Von nun an, wird die Sache komplizierter, da die geometrische Vorstellungskraft einigermaßen strapaziert wird. (Wir wollen gar nicht davon reden, wie wir vorgehen sollen, wenn wir statt eines Problems im $\mathbb{R}^3$ eines im — sagen wir — R^{1834} lösen wollen.) Es wäre also vorteilhaft, eine Formulierung des Problems zu finden, die *nicht* von der geometrischen Intuition abhängig ist.

Zu diesem Zweck führt man künstliche Variable ein und konvertiert gleichzeitig die Ungleichungen in Gleichungen. Um genauer zu sein, führt man für jede Ungleichung eine zusaätzliche Variable ein, die *Schlupfvariable* genannt wird. So sieht man, dass das ursprüngliche Problem zum Problem, die Funktion

$$\psi(x) = -350x_1 - 260x_2 - 100x_3$$

unter den Bedingungen

$$3x_1 + x_3 + x_4 = 30 \tag{1.9}$$

$$3x_1 + 6x_2 + 2x_3 + x_5 = 120 \tag{1.10}$$

$$10x_1 + 5x_2 + x_3 + x_6 = 120 \tag{1.11}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0, \tag{1.12}$$

zu minimieren, äquivalent ist. (Sollte es auch Variable x_i geben, die *sowohl* positiv *als auch* negativ sein können, ersetzt man eine solche Variable durch $x_i' - x_i''$ mit $x_i' \geq 0$ und $x_i'' \geq 0$.)

Im Allgemeinen erhält man so ein Problem des folgenden Typs: Man minimiere

$$\psi(x) = c_1x_1 + \cdots + c_nx_n$$

unter den Bedingungen

$$\begin{array}{l} a_{11}x_1 + \cdots + a_{1n}x_n + x_{n+1} = b_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n + \quad \cdots \quad + x_{n+m} = b_m. \end{array}$$

Um auf unser Beispiel zurückzukommen: Die Form (1.9)–(1.12) erlaubt es uns offenbar, leicht zu sehen, wie man entlang von Kanten ausgehend von $(0, 0, 0)$, oder genauer: $(0, 0, 0, 30, 120, 120)$ (mit den neuen Variablen!), sich bewegt. Die Form (1.9)–(1.12) ist offenbar in einer Art „Normalform“ für den Eckpunkt $(0, 0, 0, 30, 120, 120)$. „Normalform“ bedeutet, dass wir eine „Diagonalform“ in Bezug auf die Variablen, die im Eckpunkt $(0, 0, 0, 30, 120, 120)$ nicht 0 sind (das sind x_4, x_5, x_6), vor uns haben: Die Gleichungen (1.9)–(1.11) sind in einer Form, wo man „bezüglich der Variablen x_4, x_5, x_6 eliminiert hat“, und diese Variablen scheinen in der Zielfunktion $\psi(x)$ nicht auf. Man

nennt die Variablen, in Bezug auf welche man eine Diagonalform vor sich hat, *Basisvariable*. Die Basisvariablen des obigen Systems sind also $x_{n+1}, x_{n+2}, \ldots, x_m$. Den entsprechende Eckpunkt des zulässigen Bereiches erhält man dadurch, dass man die übrigen Variablen (die Nichtbasisvariablen) auf 0 setzt. Im obigen System ergibt sich der Eckpunkt

$$(0, \ldots, 0, b_1, b_2, \ldots, b_m).$$

In unserem Beispiel erhält man die Normalform, die dem Eckpunkt $(10, 0, 0, 0, 90, 20)$ entspricht, indem man x_1 durch die Nichtbasisvariablen ausdrückt und anschließend x_1 in den anderen Gleichungen eliminiert:

$$x_1 + \tfrac{1}{3}x_3 + \tfrac{1}{3}x_4 = 10 \tag{1.13}$$

$$6x_2 + x_3 - x_4 + x_5 = 90 \tag{1.14}$$

$$5x_2 - \tfrac{7}{3}x_3 - \tfrac{10}{3}x_4 + x_6 = 20 \tag{1.15}$$

und schließlich auch in der Zielfunktion:

$$\psi(x) = -350(10 - \tfrac{1}{3}x_3 - \tfrac{1}{3}x_4) - 260x_2 - 100x_3 = -260x_2 + \tfrac{50}{3}x_3 + \tfrac{350}{3}x_4 - 3500.$$

Aus dieser Normalform kann man ablesen, dass sich die Zielfunktion weiter verkleinern wird, wenn man x_2 vergrößert. Man kann außerdem aus den Gleichungen (1.13)–(1.15) ablesen, dass man x_2 bis $\min\{\frac{90}{6}, \frac{20}{5}\} = 4$ vergrößern kann. Solcherart erreicht man den Eckpunkt $(10, 4, 0, 0, 66, 0)$.

Konkrete Realisierung des Simplex–Verfahrens

Eine gebräuchliche Realisierung geschieht mit Hilfe einer Abfolge mehrerer gleichartiger „Rechentableaus“ (siehe das Beispiel weiter unten).

Gegeben sei eine Optimierungsaufgabe in der Gestalt

$$\begin{gathered} A_{11}x_1 + \cdots + A_{1N}x_N = b_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ A_{M1}x_1 + \cdots + A_{MN}x_N = b_M \\ x_1, x_2, \ldots, x_N \geq 0, \\ \psi(x) = c_1x_1 + \cdots + c_Nx_N \ldots \min. \end{gathered}$$

und eine Basis $J = \{j_1, \ldots, j_M\}$. Man löst in einem vorbereitenden Schritt (siehe Phase I) das Gleichungssystem nach $x_{j_1}, \ldots, x_{j_M}$ auf, und eliminiert in der Zielfunktion die *Basisvariablen* $x_{j_1}, \ldots, x_{j_M}$. Nun hat man die *Normalform* der Optimierungsaufgabe in Bezug auf die Basis J betimmt und kann mit dem Algorithmus beginnen.

Teilschritt 1: Eintragen in das erste Rechentableau.

BV	x_1 x_2 ... x_N	x_0	Q	$-\uparrow$	
x_{j_1}		$\bar{b}_1$			
x_{j_2}	Koeffizienten der	$\bar{b}_2$			
$\vdots$	Nebenbedingungen				
x_{j_M}		$\bar{b}_M$			
ψ	Koeff. von ψ	0			

Jedes Tableau enthält $M+1$ Zeilen. Die ersten M Zeilen sind für die Nebenbedingungen vorgesehen, die $(M+1)$-te Zeile heißt *Grundzeile* und ist für die Zielfunktion bestimmt. In der 1. Spalte stehen die Basisvariablen $x_{j_1}, \ldots, x_{j_M}$ des Eckpunktes. Die weiteren N Spalten enthalten die Koeffizienten der Variablen x_i, $i = 1, \ldots, N$ des Gleichungssystems der Nebenbedingungen und die Koeffizienten der Zielfunktion (bei Maximumsaufgabe: die *negativen* Koeffizienten). In der folgenden Spalte (mit x_0 beschriftet) sind die Werte $\bar{b}_i$ der Basisvariablen x_{j_i} und der jeweilige negative Wert (bei Maximumsaufgabe: positive Wert) der Zielfunktion für den betrachteten Eckpunkt angegeben. Es schließt sich eine Hilfsrechenspalte an, die mit Q (für „Quotient“) beschriftet ist und „Q-Spalte“ genannt wird (das entsprechende Feld der Grundzeile bleibt leer). Die letzte Spalte ist ebenfalls eine Hilfsrechenspalte und wird mit „$-\uparrow$-Spalte“ bezeichnet.

(a) Eintragen der Basisvariablen des vorgegebenen Eckpunktes in die Spalte BV.
(b) Eintragen der Koeffizienten der Nebenbedingungen und der Koeffizienten der Zielfunktion (bei Maximumsaufgabe: *negative* Koeffizienten) in die Spalten $x_1, x_2, \ldots, x_N$.
(c) Eintragen der Werte der Basisvariablen und des negativen Wertes (bei Maximumaufgaben: positiven Wertes) für den gegebenen Eckpunkt in die Spalte x_0.

Teilschritt 2: Unterstreiche die Koeffizienten der Basisvariablen in der Grundzeile. (Die unterstrichenen Zahlen müssen alle gleich 0 sein — Basisvariablen sind ja in der Zielfunktion nicht vorhanden!).

Entscheidung 1: Sind alle nicht unterstrichenen Zahlen der Grundzeile nichtnegativ (≥ 0)?

Wenn ja: Stop I: Die Optimallösung ist erreicht! Die Werte der Basisvariablen und der negative minimale Wert (bei Maximumsaufgabe: der maximale Wert) der Zielfunktion können in der Spalte x_0 abgelesen werden.

Wenn nein:

Teilschritt 3: Suche unter den nicht unterstrichenen Zahlen der Grundzeile die kleinste negative Zahl und setze dort das Merkzeichen $\uparrow$. Die im Spaltenkopf stehende Variable heißt *Eingangsvariable* und ist die neue Basisvariable. Die entsprechende Spalte heißt *Pfeilspalte*. Wird die kleinste negative Zahl in der Grundzeile an mehreren Stellen angenommen, kann eine willkürlich ausgewählt werden.

Entscheidung 2: Enthält die Pfeilspalte positive Zahlen?

Wenn nein: Stop II: Die Aufgabe ist unlösbar (beliebig kleine — beziehungsweise große für Maximumsaufgaben — Werte der Zielfunktion!).

Wenn ja:

Teilschritt 4: Bilde für alle positiven Zahlen der Pfeilspalte den Quotienten

$$\frac{\text{Zahl in der } x_0\text{-Spalte}}{\text{positive Zahl in der Pfeilspalte}}$$

und trage ihn in die Q-Spalte ein. Die Stellen der Q-Spalte, denen nichtpositive Zahlen in der Pfeilspalte entsprechen, bleiben leer.

Teilschritt 5: Bestimme das minimale Element der Q-Spalte. Die diese kleinste Zahl der Q-Spalte enthaltende Zeile heißt *Hauptzeile* und wird durch das Merkzeichen $\leftarrow$ gekennzeichnet. Die im Zeilenanfang der Hauptzeile befindliche Variable heißt

Ausgangsvariable und ist die neue Nichtbasisvariable. Die in der Pfeilspalte stehende Zahl der Hauptzeile wird durch einen Kreis markiert.

Teilschritt 6: Trage die mit (-1) multiplizierten Zahlen der Pfeilspalte in die $-\uparrow$-Spalte ein. Das entsprechende Feld der Hauptzeile bleibt leer.

Teilschritt 7: Trage die neuen Basisvariablen in die Spalte BV des neuen Rechentableaus ein. Die Basisvariablen des vorangegangenen Tableaus werden in derselben Reihenfolge eingetragen. Lediglich die Ausgangsvariable (siehe Teilschritt 5) wird durch die Eingangsvariable (siehe Teilschritt 3) ersetzt. Die Zeile des neuen Tableaus, welche die Eingangsvariable enthält, bekommt das Merkzeichen $\rightarrow$ und heißt *Hilfsrechenzeile* für das vorhergehende Tableau.

Teilschritt 8: Dividiere die Hauptzeile (Spalten $x_1, \ldots, x_N, x_0$) durch das mit einem Kreis markierte Element und trage die Ergebnisse in die Hilfsrechenzeile im folgenden Tableau ein.

Teilschritt 9: Komplettiere das nächste Rechentableau. Zur Berechnung eines Elementes $\tilde{d}_{ij}$, das nicht in der Hilfsrechenzeile steht, ist zum Element d_{ij} (d.h. zum entsprechenden Element des ursprünglichen Tableaus) das Produkt aus dem in der i-ten Zeile stehenden Element der $-\uparrow$-Spalte mit der in der j-ten Spalte stehenden Zahl der Hilfsrechenzeile zu addieren. Nach Ausführung von Teilschritt 9 ist ein Simplexschritt abgeschlossen und das neue Rechentableau ist dann mit Ausnahme der Q-Spalte und $-\uparrow$-Spalte ausgefüllt. Der nächste Simplexschritt beginnt wieder mit Teilschritt 2.

Das Ganze an unserem konkreten Beispiel durchgeführt, sieht so aus:

Zunächst müssen Schlupfvariable x_4, x_5, x_6 eingeführt werden, um aus den Ungleichungen Gleichungen zu machen. Das führt zu der folgenden Optimierungsaufgabe:

$$\begin{aligned} 3x_1 \quad\quad + \; x_3 + x_4 \quad\quad\quad &= 30 \\ 3x_1 + 6x_2 + 2x_3 \quad + x_5 \quad\quad &= 120 \\ 10x_1 + 5x_2 + \; x_3 \quad\quad + x_6 &= 120 \end{aligned}$$

mit Zielfunktion $\psi(x) = -350x_1 - 250x_2 - 100x_3 \ldots \min$. Ein Eckpunkt ist $(0, 0, 0, 30, 120, 120)$. Das Gleichungssystem ist bereits nach den zugehörigen Basisvariablen x_4, x_5, x_6 aufgelöst, auch in der Zielfunktion sind x_4, x_5, x_6 eliminiert. Daher kann bereits in das erste Tableau eingetragen werden. Die weiteren Schritte laufen stur nach den Vorschriften ab.

BV	x_1	x_2	x_3	x_4	x_5	x_6	x_0	Q	$-\uparrow$
$\leftarrow x_4$	③	0	1	1	0	0	30	10	
x_5	3	6	2	0	1	0	120	40	-3
x_6	10	5	1	0	0	1	120	12	-10
ψ	$-350\uparrow$	-260	-100	$\underline{0}$	$\underline{0}$	$\underline{0}$	0		350

BV	x_1	x_2	x_3	x_4	x_5	x_6	x_0	Q	$-\uparrow$
$\rightarrow x_1$	1	0	$\frac{1}{3}$	$\frac{1}{3}$	0	0	10	∞	0
x_5	0	6	1	-1	1	0	90	15	-6
$\leftarrow x_6$	0	⑤	$-\frac{7}{3}$	$-\frac{10}{3}$	0	1	20	4	
ψ	$\underline{0}$	$-260\uparrow$	$\frac{50}{3}$	$\frac{350}{3}$	$\underline{0}$	$\underline{0}$	3500		260

BV	x_1	x_2	x_3	x_4	x_5	x_6	x_0	Q	$-\uparrow$
x_1	1	0	$\frac{1}{3}$	$\frac{1}{3}$	0	0	10	30	$-\frac{1}{3}$
$\leftarrow x_5$	0	0	⑲/5	3	1	$-\frac{6}{5}$	66	$\frac{350}{19}$	
$\rightarrow x_2$	0	1	$-\frac{7}{15}$	$-\frac{2}{3}$	0	$\frac{1}{5}$	4	$-$	$\frac{7}{15}$
ψ	$\underline{0}$	$\underline{0}$	$-\frac{314}{3}$	$-\frac{170}{3}$	$\underline{0}$	260	4540		$\frac{314}{3}$

BV	x_1	x_2	x_3	x_4	x_5	x_6	x_0	Q	$-\uparrow$
x_1	1	0	0	$\frac{4}{57}$	$-\frac{5}{57}$	$\frac{2}{19}$	$\frac{80}{19}$		
$\rightarrow x_3$	0	0	1	$\frac{15}{19}$	$\frac{5}{19}$	$-\frac{6}{19}$	$\frac{330}{19}$		
x_2	0	1	0	$-\frac{17}{57}$	$\frac{7}{57}$	$\frac{1}{19}$	$\frac{230}{19}$		
ψ	$\underline{0}$	$\underline{0}$	$\underline{0}$	$\frac{1480}{57}$	$\frac{1570}{57}$	$\frac{4312}{19}$	$\frac{120800}{19}$		

In diesem letzten Tableau sind nun alle Koeffizienten in der Zielfunkion (der ψ-Zeile), die in einer x_i-Spalte, $i \geq 1$, stehen, nichtnegativ. Wir haben daher das Optimum gefunden! Falls man nämlich x_4, x_5 oder x_6 (die im Moment auf 0 gesetzt sind), vergrößern würde, dann würde die Zielfunktion *größer* werden. (Zur Erinnerung: Wir wollen das Minimum der Zielfunktion finden.) Daher müssen wir tatsächlich $x_4 = x_5 = x_6 = 0$ wählen, um das Minimum von $\Psi(x)$ zu erzielen, und dann ergibt sich aus den Gleichungen, dass

$$x_1 = \frac{80}{19}, \quad x_2 = \frac{230}{19}, \quad x_3 = \frac{330}{19}.$$

Der Wert des Minimums ist in der x_0-Spalte der Ψ-Zeile ablesbar: $\dfrac{120800}{19}$.

KAPITEL 2

Kombinatorische Optimierung: Das Problem des kürzesten Weges

In der *kombinatorischen Optimierung* werden Optimierungsprobleme behandelt, die in kombinatorischen Zusammenhängen auftreten. Die meisten davon sind solche, die eine bestimmte Größe, oder auch eine bestimmte Struktur, in einem Netzwerk optimieren wollen. Ich möchte ein solches Problem hier besprechen, nämlich jenes, das bei *Routenplanung* (Finden der schnellsten Zugverbindung zwischen zwei Stationen, Finden der billigsten Flugverbindung zwischen zwei Städten, usw.) entsteht: das Problem des kürzesten (billigsten, schnellsten) Weges zwischen zwei Knoten in einem Netzwerk.

Dazu müssen wir uns zuallererst darauf verständigen, was ein Netzwerk ist. Der mathematisch Begriff, der ein Netzwerk „modelliert“ ist der Begriff des (*kombinatorischen*) *Graphen*.

DEFINITION. Ein *Graph* ist ein Paar[1] (V, E), wobei V die *Menge der Knoten* ist, und E die *Menge der Kanten*,

$$E \subseteq \{\{x, y\} : x, y \in V \text{ und } x \neq y\}.$$

Die bildliche Realisierung eines Graphen sieht so wie in Figur 1 aus: Knoten werden als dicke Punkte dargestellt, während eine Kante die beiden Knoten x, y, aus denen sie besteht, verbindet. Wir werden später oft xy (oder yx) für die Kante, die x und y verbindet (so sie existiert) schreiben. Der Graph in Figur 1 besteht aus $V = \{a, b, c, d, e, f, u, v\}$ und

$$E = \{\{u, a\},\ \{u, d\},\ \{a, b\},\ \{a, c\},\ \{d, c\},\ \{d, f\},\ \{b, v\},\ \{c, e\},\ \{f, e\},\ \{e, v\}\}.$$

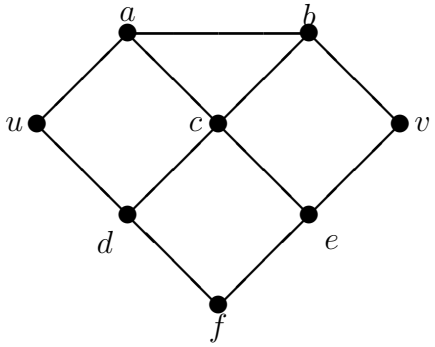

ABBILDUNG 1. Ein Graph

[1]Die Buchstaben V und E für Knoten- beziehungsweise Kantenmenge sind in Anlehnung an die englischen Begriffe *“vertex”* (für *Knoten*) und *“edge”* (für *Kante*) heutzutage gebräuchlich.

Die Knoten im Graphen entsprechen etwa Stationen in einem Zugverbindungsnetzwerk oder Städten, und die Kanten entsprechen Zugsverbindungen oder Flugverbindungen. Jeder Kante muss noch eine Zahl zugeordnet werden, die dann der Länge einer Verbindung, der Zeit, die man benötigt, um von einem Knoten („Station“) zu einem anderen zu gelangen, oder dem Preis entspricht, den ich bezahlen muss, um einen bestimmten Flug zu kaufen. Wir führen also eine Funktion $w : E \to \mathbb{R}_{\geq 0}$ von den Kanten in die nichtnegativen reellen Zahlen ein, die diese genannten Informationen (Länge, Zeit, Preis, usw.) wiedergibt. Wir nennen sie *Gewichtsfunktion*. In Figur 2 ist diese Funktion durch die Zahlen angedeutet, die bei den Kanten des Graphen stehen.

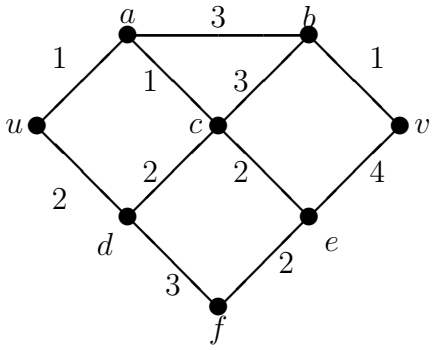

ABBILDUNG 2. Ein Graph mit Gewichtsfunktion

Die Aufgabe ist es nun die kürzeste (schnellste, billigste) Verbindung zwischen zwei gegebenen Knoten u und v zu finden, also einen Weg

$$uu_1u_2 \dots u_mv$$

(hier sind $u_1, u_2, \dots, u_m \in V$, alle paarweise verschieden, und uu_1, u_1u_2, …, u_mv sind Kanten in E), der die Eigenschaft hat, dass

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_mv)$$

(die *„Länge“* des Weges von u nach v) minimal ist.

Der Algorithmus von Dijkstra

Edskar W. Dijkstra hat einen Algorithmus 1956 vorgeschlagen, der dieses Problem effizient löst. In diesem Algorithmus startet man mit der Menge $Q = \{u\}$. Anschließend sieht man sich sukzessive benachbarte[2] Knoten an und nimmt (“greedy– „gierig“) denjenigen zur Menge Q hinzu, der darunter den kürzesten Abstand zu u hat. Dieser Abstand eines Knoten $x \in Q$ zu u wird in einer Funktion $f : Q \to V \times \mathbb{R}_{\geq 0}$ gespeichert, die außerdem auch als zusätzliche Information den Vorgängerknoten auf einem kürzesten Weg von u nach x enthält.

Soweit so unverständlich. Hier ist die genaue Beschreibung dieses Algorithmus.

INITIALISIERUNG: $Q := \{u\}$, $f(u) := (-, 0)$.

[2]Zwei Knoten in einem Graphen, die durch eine Kante verbunden sind, heißen *benachbart*, und man nennt sie *Nachbarn*.

WIEDERHOLUNGSSCHRITT: Betrachte alle Knoten $x \in V \backslash Q$, die zu einem Knoten in Q, sagen wir $y \in Q$, benachbart sind. Für alle diese Knoten x (und Nachbarn y) berechnet man

$$f(y)_2 + w(yx). \tag{2.1}$$

(Hier bezeichnet $f(y)_2$ die zweite Komponente im Paar $f(y)$.) Unter allen diesen Werten bestimmen wir das Minimum. Sei $x_0 \in V \backslash Q$ ein solcher Knoten, der dieses Minimum zusammen mit seinem Nachbarn y_0 erzielt. Wir setzen nun

$$Q := Q \cup \{x_0\}, \quad f(x_0) = (y_0, f(y_0)_2 + w(y_0x_0)).$$

Falls $v \notin Q$, dann führt man den Wiederholungsschritt noch einmal — mit dem neuen Q und erweiterten Definition der Funktion f — aus. Falls $v \in Q$, dann geht man zum nächsten Schritt.

AUSGABE DES ERGEBNISSES: Die kürzeste Länge eines Weges von u nach v ist gleich $f(v)_2$, und der Weg lässt sich aus den ersten Komponenten der Funktion f auf naheliegende Art rekonstruieren.

Zur Illustration wenden wir diesen Algorithmus nun auf unser Beispiel In Figur 2 an.

Wir beginnen mit der Initialisierung:

$$Q := \{u\}, \quad f(u) := (-, 0).$$

Es folgt die (erste) Anwendung des Wiederholungsschrittes. Die Nachbarn von u sind a und d. Wir berechnen

$$f(u)_2 + w(ua) = 0 + 1 = 1,$$
$$f(u)_2 + w(ud) = 0 + 2 = 2.$$

Das Minimum ist die erste der beiden Zahlen, nämlich 1. Wir setzen also

$$Q := \{u, a\}, \quad f(a) := (u, 1). \tag{2.2}$$

Der Knoten v ist (noch) nicht in Q. Wir müssen daher den Wiederholungsschritt noch einmal durchführen, mit dem neuen Q. Die Nachbarn der Knoten in Q (die noch nicht in Q enthalten sind) sind b, c, d. Wir berechnen

$$f(a)_2 + w(ab) = 1 + 3 = 4,$$
$$f(a)_2 + w(ac) = 1 + 1 = 2,$$
$$f(u)_2 + w(ud) = 0 + 2 = 2.$$

Das Minimum ist 2, das zweimal auftritt. Wählen wir die erste Möglichkeit (den Knoten c). Wir setzen

$$Q := \{u, a, c\}, \quad f(c) := (a, 2).$$

Der Knoten v ist nicht in Q. Wir führen den Wiederholungsschritt wieder durch. Die Nachbarn der Knoten in Q (die noch nicht in Q enthalten sind) sind b, e, d. Wir berechnen

$$\begin{aligned} f(a)_2 + w(ab) &= 1 + 3 = 4, \\ f(c)_2 + w(cb) &= 2 + 3 = 5, \\ f(c)_2 + w(ce) &= 2 + 2 = 4, \\ f(u)_2 + w(ud) &= 0 + 2 = 2, \\ f(c)_2 + w(cd) &= 2 + 2 = 4. \end{aligned}$$

Das Minimum ist 2. Wir setzen

$$Q := \{u, a, c, d\}, \quad f(d) := (u, 2).$$

Der Knoten v ist nicht in Q. Wir führen den Wiederholungsschritt wieder durch. Die Nachbarn der Knoten in Q (die noch nicht in Q enthalten sind) sind b, e, f. Wir berechnen

$$\begin{aligned} f(a)_2 + w(ab) &= 1 + 3 = 4, \\ f(c)_2 + w(cb) &= 2 + 3 = 5, \\ f(c)_2 + w(ce) &= 2 + 2 = 4, \\ f(d)_2 + w(df) &= 2 + 3 = 5. \end{aligned}$$

Das Minimum ist 4. Wir wählen die ersten Möglichkeit und setzen

$$Q := \{u, a, c, d, b\}, \quad f(b) := (a, 4). \tag{2.3}$$

Der Knoten v ist nicht in Q. Wir führen den Wiederholungsschritt wieder durch. Die Nachbarn der Knoten in Q (die noch nicht in Q enthalten sind) sind v, e, f. Wir berechnen

$$\begin{aligned} f(b)_2 + w(bv) &= 4 + 1 = 5, \\ f(c)_2 + w(ce) &= 2 + 2 = 4, \\ f(d)_2 + w(df) &= 2 + 3 = 5. \end{aligned}$$

Das Minimum ist 4. Wir setzen

$$Q := \{u, a, c, d, b, e\}, \quad f(e) := (c, 4).$$

Der Knoten v ist nicht in Q. Wir führen den Wiederholungsschritt wieder durch. Die Nachbarn der Knoten in Q (die noch nicht in Q enthalten sind) sind v, f. Wir berechnen

$$\begin{aligned} f(b)_2 + w(bv) &= 4 + 1 = 5, \\ f(e)_2 + w(ev) &= 4 + 4 = 8, \\ f(d)_2 + w(df) &= 2 + 3 = 5, \\ f(e)_2 + w(ef) &= 4 + 2 = 6. \end{aligned}$$

Das Minimum ist 5. Klarerweise wählen wir die erste Mögl;ichkeit und setzen

$$Q := \{u, a, c, d, b, e, v\}, \quad f(v) := (b, 5). \tag{2.4}$$

Wir sind am Ziel! Der Knoten v ist in Q! Die minimale Länge eines Weges von u nach v ist 5, und einen kürzesten Weg findet man über die ersten Komponenten der Funktion f in (2.4), (2.3) und (2.2). nämlich u, a, b, v.

Warum funktioniert der Algorithmus? Dies folgt aus der folgenden Behauptung, die eingangs als Idee des Algorithmus formuliert wurde.

BEHAUPTUNG 2.1. *Während der Ausführung des Algorithmus gilt zu jedem Zeitpunkt: Für einen Knoten $x \in Q$ gibt $f(x)_2$ die Länge eines kürzesten Weges von u nach x an, und $f(x)_1$ den Vorgängerknoten von x auf einem solchen kürzesten Weg.*

BEWEIS. Wir beweisen die Behauptung mit Induktion nach der Anzahl der Wiederholungsschritte im Algorithmus. Ganz am Anfang ist die Behauptung offensichtlich richtig, denn $f(u)_2 = 0$. (Die Länge eines kürzesten Weges von u nach u ist in der Tat gleich 0).

Wir nehmen nun an, dass die Behauptung zu einem bestimmten Zeitpunkt während der Ausführung des Algorithmus richtig ist.

Für den Induktionsschritt müssen wir beweisen, dass der Knoten x_0, der nun im Wiederholungsschritt zu Q hinzugefügt wird, ebenfalls die Behauptung erfüllt. Sei der Nachbarknoten von x_0, der das Minimum in (2.1) erzielt, der Knoten $y_0 \in Q$.

Sei nun

$$uu_1u_2 \dots u_m x$$

ein Weg von u nach x. Wir müssen zeigen, dass die Länge

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_m x)$$

dieses Weges mindestens $f(y_0)_2 + w(y_0x_0)$ ist.

Von den Knoten $u_1, u_2, \dots, u_{m-1}$ sind die ersten paar in Q. Sei k minimal, sodass $u_k \in Q$, aber $u_{k+1} \notin Q$. Es gilt dann

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_m x) \geq w(uu_1) + w(u_1u_2) + \cdots + w(u_{k-1}u_k) + w(u_k u_{k+1}).$$

Gemäß Induktionsvoraussetzung wissen wir, dass

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_{k-1}u_k) = f(u_k)_2.$$

Wir erhalten so

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_m x) \geq f(u_k)_2 + w(u_k u_{k+1}).$$

Die rechte Seite ist eine Zahl, die in (2.1) bei Ausführung des Wiederholungsschrittes betrachtet wird. Wir wissen, dass das Minimum aller dieser Zahlen gleich $f(y_0)+w(y_0x_0)$ ist. In anderen Worten, wir haben tatsächlich

$$w(uu_1) + w(u_1u_2) + \cdots + w(u_m x) \geq f(y_0) + w(y_0x_0)$$

bewiesen. □

KAPITEL 3

Suchprobleme — Elemente der Informationatheorie

Suchprobleme treten im täglichen Leben ständig — explizit und implizit — auf: Internetsuche, Datenbanksuche, selbst viele Algorithmen, die standardmäßig in Computerprogrammen ablaufen — wie etwa Sortieralgorithmen — sind allesamt Suchprobleme. Ich möchte hier die fundamentale Idee, die am Beginn (der Informationstheorie) steht, besprechen: die informationstheoretische Schranke.

Sehr vereinfacht[1] ist eine Suchabfrage in einer Datenbank vergleichbar mit dem folgenden Spiel: Eine Person denkt sich eine ganze Zahl zwischen 1 und 100 aus, und es ist nun Aufgabe der anderen Mitspieler(innen) möglichst rasch mit Entscheidungsfragen (Fragen, die mit „ja“ oder „nein“ zu beantworten sind) die ausgedachte Zahl zu erraten.[2]

Eine (naive) Strategie würde darin bestehen, auf gut Glück zu fragen: *„Ist es die Zahl 43?“* Wenn wir Glück haben, dann haben wir unser Ziel erreicht: Wir haben die Zahl erraten. Falls nicht, probieren wir die nächste Zahl, usw. Das ist aber keine allzu intelligente Strategie, da ich — wenn ich Pech habe — sehr, sehr oft fragen muss, bis ich die Zahl erraten habe. (Im schlechtesten Fall muss ich 100-mal fragen!)

Geht es besser? Eine andere Strategie besteht darin, den Zahlraum, wo sich die gesuchte Zahl befindet, mit meinen Fragen möglichst einzuschränken. Ich frage: „Ist die Zahl ≤ 50?“ Je nachdem, wie die Antwort ausfällt, weiß ich danach, dass sich die gesuchte Zahl entweder im Bereich 1 bis 50 befindet oder im Bereich 51 bis 100. Mit anderen Worten: Nach dieser ersten Frage habe ich die Suche auf $50 = 100/2$ Zahlen eingeschränkt.

Ich kann dann diese Strategie der „Halbierung“ fortsetzen: Nehmen wir an, dass die Antwort auf die erste Frage „nein“ war. Dann weiß ich nun, dass sich die gesuchte Zahl im Bereich $[51, 100]$ befindet. Ich frage nun: „Ist die Zahl ≤ 75?“ Je nachdem, wie die Antwort ausfällt, habe ich den Suchbereich nun auf $50/2 = 25$ Zahlen eingeschränkt.

So fährt man fort. Nehmen wir an, dass die Antwort auf die zweite Frage „ja“ war. Damit weiß ich, dass sich die gesuchte Zahl im Bereich $[51, 75]$ befindet. Ich frage nun: „Ist die Zahl ≤ 62?“ Das halbiert den verbleibenden Suchraum zwar nicht exakt (die Hälfte von 25 ist keine ganze Zahl), aber es halbiert ihn „so gut wie eben möglich“. (Es teilt ihn in $25 = 12 + 13$.)

[1] Wenn man tiefer gehen möchte, dann müsste man auch in Betracht ziehen, wie oft im Durchschnitt bestimmte Stichwörter abgefragt werden, und weiter ins Detail gehende Aspekte. Das würde uns dann tiefer in die Informationstheorie und insbesondere zu Claude Shannon's erstem Hauptsatz der Informationstheorie führen.

[2] Die Zahlen in diesem Spiel entsprechen den Stichwörtern in der Datenbank, die ausgedachte Zahl dem Stichwort, das in der Datenbank gesucht wird, und die Fragen entsprechen den Abfragen, die im Computerprogramm ablaufen, das das eingegebene Stichwort in der Datenbank sucht und schließlich findet.

Je nachdem, wie die Antwort auf die dritte Antwort ausfällt, fährt man im selben Stil fort. Schlussendlich wird man nach spätestens[3] $\lceil \log_2 100 \rceil = \lceil 6.643\ldots \rceil = 7$ Fragen die gesuchte Zahl erraten haben.

Geht es noch besser? Der folgende Satz über die informationstheoretische Schranke besagt, dass es (im schlechtesten Fall) nicht besser geht.[4]

SATZ 3.1. *In einem Suchraum mit n Elementen suchen wir ein bestimmtes Element. Erlaubte Tests/Fragen sind solche, die genau zwei Antworten („ja/nein“) zulassen. Dann benötigt man im schlechtesten Fall mindestens $\lceil \log_2 n \rceil$ Fragen, bis man das gesuchte Element gefunden hat.*

BEWEIS. Wir führen den Beweis mittels Induktion nach n.

Für $n = 1$ ist die Aussage des Satzes trivial: Einerseits benötigen wir natürlich keine Frage, um dieses eine Element im Suchraum zu finden. Andererseits gilt $\log_2 1 = 0$.

Nehmen wir also nun an, dass die Aussage des Satzes für alle Suchräume mit weniger als n Elemente gilt. Wir stellen nun also eine Suchabfrage an unseren Suchraum mit n Elementen. Je nachdem, wie die Antwort ausfällt, haben wir die verbleibenden Möglichkeiten auf a oder auf b Elemente eingeschränkt, wobei $a + b = n$ ist.[5] Eine der beiden Zahlen a, b muss $\geq n/2$ sein. Ohne Beschränkung der Allgemeinheit sei diese Zahl a. Laut Induktionsannahme brauchen wir nun noch (im schlechtesten Fall) $\lceil \log_2 a \rceil \geq \lceil \log_2(n/2) \rceil = \lceil \log_2 n \rceil - 1$ Fragen. Zusammen mit der allerersten Frage sind das insgesamt also mindestens $(\lceil \log_2 n \rceil - 1) + 1 = \lceil \log_2 n \rceil$ Fragen, wie behauptet. □

[3]Hier bezeichnet $\lceil x \rceil$ die kleinste ganze Zahl, die $\geq x$ ist. Der *Logarithmus von x zur Basis* 2, $\log_2 x$, ist jene Zahl α, für die $2^\alpha = x$ gilt.

[4]Der Satz könnte noch auf Tests/Fragen, die mehr als zwei Antworten zulassen, verallgemeinert werden. Lässt man Tests/Fragen zu, die q Antworten erlauben, dann kann man die Schranke im Satz durch $\lceil \log_q n \rceil$ ersetzen.

[5]Wir können annehmen, dass sowohl a als auch b kleiner als n sind. Andernfalls hatte ich eine sinnlose Frage gestellt, die mich überhaupt nicht weiterbringt.

KAPITEL 4

Kryptographie: RSA-Verschlüsselung

Wir kommen zu einer Anwendung der Mathematik in der Datenübertragung. Ich werde hier nicht auf das (in der Einführung erwähnte) Problem, Datenübertragung trotz imperfekter Übertragung fehlerfrei durchführen zu können, eingehen, aber auf das andere Problem: die *Sicherheit* von Datenübertragung gegenüber von „Spionen".

Heutzutage bewegen wir uns alle im Internet und wickeln dabei immer wieder Dinge ab, die eine „sichere" Datenübertragung benötigen, wie etwa das Eingeben eines privaten Passwortes, der eigenen Bankdaten, der eigenen Kreditkartennummer, usw. Wir alle haben wohl schon davon gehört, dass hier die sogenannte *RSA-Verschlüsselung* zur Anwendung kommt, und dass das etwas mit großen Primzahlen zu tun hat. „RSA" steht für Ron **R**ivest, Adi **S**hamir und Leonard **A**dleman, die 1977 ein Verschlüsselungssystem vorgeschlagen haben, das sich (jedenfalls bis zum heutigen Tag) als sehr robust gegen Angriffe von außen herausgestellt hat. Dieses werde ich im Folgenden erklären. Wir brauchen dazu nichts außer ein wenig elementare Zahlentheorie.

Es geht um folgende Situation: Ein Anbieter auf dem Internet (eine Firma, eine Bank, eine Bibliothek, usw.) möchte es Kunden möglich machen, seine Dienste zu nützen — die allerdings nicht frei zugänglich sind. Dafür ist es notwendig, dass der Kunde ein Passwort eingibt, seine Kreditkartennummer, oder Ähnliches. Aus Sicht des Anbieters sind es also *alle anderen*, die eventuell Kunden sein könnten. Die *Methode* der Verschlüsselung muss also *jedem* zugänglich sein (die Spione, Betrüger, usw. inklusive ...). Die *Entschlüsselung* darf aber nur dem *Anbieter* (leicht) möglich sein (in keinem Fall den Spionen, Betrügern, ...). Klingt beinahe unmöglich, oder?

Um diese Aufgabe doch zu bewältigen, wählt man zwei ungefähr gleich große (aber sehr, sehr große) verschiedene Primzahlen p und q. Um ein Beispiel zur Illustration bei der Hand zu haben, wählen wir $p = 11$ und $q = 13$.[1]

Der „Öffentlichkeit" wird der *öffentliche Schlüssel* (*"public key"*) (M, e) bekannt gegeben, wobei $M = pq$ und e eine natürliche Zahl ist, die teilerfremd zu $(p-1)(q-1)$ ist. Der Anbieter verfügt über den geheimen *privaten Schlüssel* (*private key*) (M, d), wobei

$$e \cdot d \equiv 1 \pmod{(p-1)(q-1)}. \tag{4.1}$$

An dieser Stelle ist es wichtig, immer daran zu denken, dass die beiden Primzahlen p und q *nicht* öffentlich bekanntgegeben werden, sondern nur deren Produkt M. In

[1] Das sind definitiv keine großen Primzahlen (und schon gar keine sehr, sehr großen ...). Sie sollen nur zur Illustration der Idee und der Methode dienen. Gebräuchliche RSA-Verschlüsselung heutzutage benutzt M's in der Größenordnung von 300 bis 500 Dezimalstellen. Da unsere Computer aber ständig mächtiger werden, und es somit mit der Zeit möglich wird, immer größere Zahlen mit Hilfe des Computers in Primfaktoren zu zerlegen, ist es immer wieder notwendig, die verwendeten Primzahlen p und q zu vergrößern.

unserem Beispiel haben wir $M = 11 \cdot 13 = 143$. Außerdem wählen wir $e = 7$ und somit $d = 103$. Für diese Wahl gilt tatsächlich

$$e \cdot d = 7 \cdot 103 = 721 \equiv 1 \pmod{120},$$

wobei $(p-1)(q-1) = 10 \cdot 12 = 120$.

Nehmen wir nun an, dass ein Kunde eine Nachricht (Passwort, Kreditkartennummer, ...) an den Anbieter senden will, dargestellt durch die ganze Zahl[2] m, die teilerfremd zu M sein soll. Er berechnet dann (lokal) m^e modulo M. Sei diese Zahl x. Die Zahl x wird dann an den Anbieter verschickt. In anderen Worten: Sowohl der Anbieter als auch der Spion wissen über die Methode, über (M, e), Bescheid, und nun erhalten sie x. Daraus müssen sie m berechnen. Angenommen, der Kunde möchte die Nachricht $m = 111$ verschicken: Er berechnet (lokal)

$$111^7 = 207616015289871 \equiv 45 \pmod{143}.$$

Er sendet also $x = 45$.

Für den Anbieter ist das Rekonstruieren von m ein Kinderspiel: Dieser kennt nämlich die Zahl d, die (4.1) erfüllt. Er muss bloß x^d modulo M berechnen und das Resultat ist m. In der Tat: Wegen (4.1) gibt es eine ganze Zahl k, sodass $e \cdot d = k(p-1)(q-1)+1$. Auf Grund von Satz 4.2 (weiter unten) gilt dann

$$x^d = (m^e)^d = m^{e \cdot d} = m^{k(p-1)(q-1)+1} = \left(m^{(p-1)(q-1)}\right)^k m^1 \equiv m \pmod{M}.$$

Im konkreten Beispiel berechnet der Anbieter

$$45^{103} = \langle \text{sehr große Zahl} \rangle \equiv 111 \pmod{143}.$$

Für den Spion schaut die Sache anders aus. Er kennt nämlich d *nicht*. Nun wird man sagen: Aber so schwer ist das auch wieder nicht. Er muss sich eben „nur“ das d aus der Kongruenz (4.1) ausrechnen. Was müsste daür getan werden? Wir zerlegen M in $p \cdot q$, berechnen $(p-1)(q-1)$ und lösen dann die Kongruenz (4.1) nach d. Die letzten zwei Schritte sind in der Tat einfach zu bewerkstelligen, nicht aber die Zerlegung von M in seine beiden Faktoren p und q (wenn ich sie eben nicht vorher kenne)! Hier liegt der springende Punkt:

> *Es ist (wahrscheinlich) rechnerisch sehr aufwendig, große Zahlen in ihre Primfaktoren zu zerlegen.*[3,4]

Unter dieser Annahme kann es einem Spion *nicht* gelingen, die allen bekannte Zahl M in $p \cdot q$ zerlegen, da unsere heutigen Computer zu lange und zu viel Speicherplatz

[2] Es wird jedem wohl einleuchten, dass man Nachrichten immer durch Zahlen darstellen kann.

[3] Dies ist eine Vermutung (die sich auch präzise formulieren lässt), die nicht bewiesen ist. In jedem Fall ist derzeit kein effizienter Algorithmus bekannt, der das Problem der *Primfaktorzerlegung* mit Hilfe eines herkömmlichen *digitalen* Computer löst. Peter Shor hat allerdings gezeigt, dass ein *Quantencomputer* — so er einmal gemäß den derzeitigen Vorstellungen existieren sollte — dieses Problem effizient lösen könnte (und somit die RSA-Verschlüsselung wertlos machen würde).

[4] Das Problem der Primfaktorzerlegung darf nicht mit dem (einfacheren) Problem der *Primzahlerkennung* verwechselt werden. Beim letzeren geht es darum, zu erkennen, ob eine gegebene Zahl eine Primzahl ist (oder nicht). Manindra Agrawal, Neeraj Kayal und Nitin Saxena haben gezeigt, dass dieses Problem effizient lösbar ist.

benötigen würden, um die Faktorisierung zu finden. Somit ist es einem Spion nicht möglich dieses Verschlüsselungssystem zu „knacken“.[5]

Im eben erklärten RSA-Verschlüsselungssystem wurde eine Tatsache verwendet, die in Satz 4.2 vorgestellt wird.[6] Zum Beweis des Satzes benötigen wir zunächst den folgenden Hilfssatz.

LEMMA 4.1. *Es seien p und q zwei verschiedene Primzahlen. Es gibt $(p-1)(q-1)$ ganze Zahlen zwischen 1 und $p \cdot q$, die nicht durch p oder q teilbar sind.*

BEWEIS. Es gibt $p \cdot q$ ganze Zahlen zwischen 1 und $p \cdot q$, q davon sind durch p teilbar, p davon sind durch q teilbar. Nur eine davon, nämlich $p \cdot q$ selbst, ist durch beide Primzahlen teilbar. Somit gibt es zwischen 1 und $p \cdot q$ genau

$$p \cdot q - p - q + 1 = (p-1)(q-1)$$

ganze Zahlen, die weder durch p noch durch q teilbar sind. □

SATZ 4.2. *Seien wiederum p und q zwei verschiedene Primzahlen, und sei m eine positive ganze Zahl die teilerfremd zu $p \cdot q$ ist. Dann gilt*

$$m^{(p-1)(q-1)} \equiv 1 \pmod{p \cdot q}.$$

BEWEIS. Bezeichnen wir die ganzen Zahlen zwischen 1 und $p \cdot q$, die teilerfremd zu $p \cdot q$ sind, mit

$$x_1, x_2, \ldots, x_{(p-1)(q-1)}.$$

Für $i \neq j$ gilt

$$m \cdot x_i \not\equiv m \cdot x_j \pmod{p \cdot q}.$$

Dies sieht man dadurch, dass man das Gegenteil annimmt. Dieses würde nämlich bedeuten, dass $p \cdot q$ die Differenz $m \cdot x_i - m \cdot x_j = m(x_i - x_j)$ teilen würde. Da — auf Grund unserer Annahmen — $p \cdot q$ teilerfremd zu m und $x_i - x_j$ ist (es gilt $|x_i - x_j| < p \cdot q$), haben wir einen Widerspruch vorliegen.

Eine Folgerung der obigen Überlegungen ist, dass

$$m \cdot x_1, m \cdot x_2, \ldots, m \cdot x_{(p-1)(q-1)} \pmod{p \cdot q} \tag{4.2}$$

alle Restklassen

$$x_1, x_2, \ldots, x_{(p-1)(q-1)} \pmod{p \cdot q} \tag{4.3}$$

durchläuft. Wenn wir also das Produkt aller Zahlen in (4.2) bilden, muss — modulo $p \cdot q$ — das selbe herauskommen, wie beim Produkt aller Zahlen in (4.3). Es gilt also

$$\prod_{i=1}^{(p-1)(q-1)} m \cdot x_i \equiv \prod_{i=1}^{(p-1)(q-1)} x_i \pmod{p \cdot q},$$

oder, nach Durchkürzen,

$$m^{(p-1)(q-1)} \equiv 1 \pmod{p \cdot q}.$$ □

[5]Um der Wahrheit die Ehre zu geben, muss man bei der Auswahl der Primzahlen p und q und bei der Wahl von e auf weitere Dinge Acht geben, um „Angriffen“ zu entgehen, die auf gewisse spezielle Konstellationen eingestellt sind. Hier ist aber nicht der Platz, um auf solche Feinheiten einzugehen.

[6]Der Satz ist ein Spezialfall eines Satzes von Leonhard Euler.