

# Die optimierte Odyssee

Das Problem der kürzesten Rundreise ist Prototyp einer großen Klasse praktisch bedeutsamer, komplexer Minimierungs- oder Maximierungsaufgaben. Sie sind so schwer, daß man sich häufig mit einer brauchbaren Näherung zufriedengeben muß. Neue, listenreiche Verfahren liefern jedoch immer häufiger die nachweislich beste Lösung.

VON MARTIN GRÖTSCHEL UND MANFRED PADBERG

**O**dyseus, der sagenhafte König von der griechischen Insel Ithaka, hatte eine lange Reise hinter sich, als er nach zwanzig Jahren seine Penelope endlich wieder in die Arme schließen konnte – Stoff für ein ganzes Epos, die „Odyssee“ des antiken Dichters Homer. Moderne Interpreten haben aus ihr 16 Orte im Mittelmeerraum herausgelesen, die er besucht haben soll (kleines Bild unten).

Er hätte es zweifellos kürzer haben können, wären da nicht das Schicksal, die Launen des Meeres, die Mißgunst der Götter und vor allem der mangelnde Überblick über die Geographie gewesen. Versetzen wir Odysseus in die moderne Zeit: Stellen wir uns einen Topmanager oder – prosaischer – Handelsvertreter (*travelling salesman*) vor. Er will alle 16 Städte in beliebiger Reihenfolge besuchen, und zwar auf dem kürzestmöglichen Wege – Zeit ist Geld. Das ist das berühmte *travelling salesman problem* (TSP): schnell formuliert, höchst praxisrelevant, scheinbar harmlos, in Wirklichkeit aber – für große Städteanzahlen – so schwer, daß die Mathematiker sich auch nach Jahrzehnten harter Arbeit die Zähne daran ausbeißen.

Mehr noch: Das TSP ist nur das einleuchtendste Beispiel aus einer ganzen Klasse von Problemen. Sie finden sich

überall da, wo es gilt, unter Kombinationen von sehr vielen Möglichkeiten die günstigste zu finden. Beispielhaft für die unüberschaubar vielen Anwendungen seien genannt:

- Zuweisung von Ladung und Personal an Lastwagen oder Frachtflugzeuge,
- Maschinenbelegung,
- Fahrplangestaltung,
- Entwurf von Mikrochips,
- Kapitalanlageplanung und
- die Gestaltung von Kommunikationsnetzen.

Für das Problem des Odysseus ergibt sich als kürzeste Rundreise eine Tour von 6859 Kilometern Länge (Kasten Seite 40 rechts unten). Die Entfernungen zwischen je zwei Städten haben wir in Luftlinie gemessen (und auf ganze Kilometer gerundet); aber

selbst mit dem Flugzeug hätte Odysseus in der von ihm gewählten Reihenfolge 9913 Kilometer zurücklegen müssen. Optimierung kann also erhebliche Einsparungen bringen!

Wie haben wir nun die kürzeste Rundreise für Odysseus gefunden? Irgendwelche Touren zu finden ist nicht schwer: Es stehen 653 837 184 000 Möglichkeiten zur Auswahl. Wir haben alle diese Touren durchprobiert und ihre Längen berechnet (Kasten auf Seite 34). Das hat auf einem üblichen Arbeitsplatzrech-

Der sagenhafte Odysseus drohte bereits auf der zweidimensionalen Meeresoberfläche rettungslos verlorenzugehen und wählte nicht den kürzesten Weg nach Hause (kleines Bild, nach der Interpretation von E. Bradley). Um nicht nur heimzufinden, sondern auf kürzestem Wege alle Stationen aufzusuchen, muß man sich statt in zwei in 120 Dimensionen zurechtfinden.



ZEICHNUNG: PETER CONNOLLY FS  
 DATEN: BRITISH ARMY  
 MEDICAL WAR OCEANIC DATA BASE (MARINE  
 SCIENCE AND TECHNOLOGY PROGRAMME BER EU)  
 MARTIN GRÖTSCHEL, MANFRED PADBERG



**Weltrekord: die kürzeste Rundreise durch sämtliche 13509 Städte der USA (ohne Alaska und Hawaii) mit mehr als 500 Einwohnern**

ner immerhin 92 Stunden – fast vier Tage – gedauert.

Diese Enumeration ist der klassische Ansatz zur Lösung von kombinatorischen Optimierungsproblemen, das heißt solchen mit endlich vielen Auswahlmöglichkeiten. Es steckt nichts dahinter als die gedankenlose Anwendung roher Gewalt; aber diese Methode lebt immer noch in einigen (etwas angestaubten) Köpfen fort. Nur scheitert sie bereits an Problemen, die geringfügig größer sind als die Reiseplanung des Odysseus. Wir haben dieses Beispiel unter anderem deswegen gewählt, weil es mit heutzutage vorhandenen Computern gerade noch mittels Enumeration lösbar ist. Bereits 25 Städte würden die größten Supercomputer mit den ausgefeiltesten Enumerationstechniken hoffnungslos überfordern.

Praktische Probleme sind jedoch deutlich größer – und werden trotzdem gelöst. Erst im Sommer 1998 haben David Applegate, Robert Bixby und William Cook von der Rice-Universität in Houston (Texas) sowie Vasek Chvatal von der Rutgers-Universität in New Brunswick (New Jersey) mit den hier beschriebenen Methoden die kürzeste Rundreise durch mehr als 13 000 Städte gefunden (Bild oben) und damit ihren bisherigen Rekord von 7397 Städten weit hinter sich gelassen. Solche und noch weit größere Probleme treten täglich in der Produktion von Leiterplatten (Kasten Seite 35), beim Entwurf höchstintegrierter Schaltkreise, in der Röntgen-

kristallographie und in vielen anderen Gebieten auf.

Am Anfang der Analyse steht eine mathematische Formalisierung. Beim TSP ersetzt man jede Stadt durch einen Punkt, einen sogenannten Knoten, und jede direkte Verbindung zwischen zwei Städten durch eine Linie (Mathematiker sagen: Kante) zwischen den entsprechenden Knoten. Ein Gebilde aus Knoten mitsamt Kanten dazwischen heißt ein Graph. Wenn, wie beim Problem des Odysseus, jeder Knoten mit jedem anderen durch eine Kante verbunden ist – man kann von jeder Stadt direkt zu jeder anderen fliegen –, sprechen die Fachleute von einem vollständigen Graphen. Der vollständige Graph mit 16 Knoten hat genau 120 Kanten.

#### **Mathematische Modellierung: Aus einer Rundreise wird ein Graph**

An jede Kante wird nun die Streckenlänge zwischen den zugehörigen Knoten angeschrieben – allgemeiner formuliert: die „Kosten“ (Zeit, Geld, Material, oder was einem sonst lieb und teuer ist), die das Durchlaufen dieser Kante verursacht. Jede Reise entspricht einer Teilmenge der Kanten. Wenn eine solche Teilmenge eine komplette Rundreise durch alle Knoten bildet, wird sie Tour oder Hamilton-Kreis genannt, nach dem irischen Mathematiker Sir William Hamilton (1805 – 1865). Die Länge einer Tour ist

die Summe der einzelnen Kantenlängen. Das TSP ist also, graphentheoretisch formuliert, die Aufgabe, die kürzeste Tour in einem vollständigen Graphen zu finden; genauer: eine kürzeste Tour, denn es kann verschiedene Touren mit der gleichen minimalen Länge geben.

Warum ist nun das TSP so schwierig? Liegt es daran, daß die Zahl der Touren für große Städtezahlen so ungeheuer groß ist? Erstaunlicherweise nein. Die Anzahl möglicher Lösungen ist kein brauchbares Maß für die Schwierigkeit eines kombinatorischen Problems.

Betrachten wir als Gegenbeispiel folgendes Problem. Zwischen den 16 Städten soll ein Kommunikationsnetzwerk errichtet werden, das jede Stadt mit jeder anderen verbindet, möglicherweise auf dem Umweg über weitere Städte. Wieder gehören zu einer Direktverbindung (Kante) zwischen zwei Städten gewisse – entfernungsabhängige – Kosten, etwa für die Verlegung eines Glasfaserkabels. Die teuerste Lösung besteht im vollständigen Graphen: Jede Stadt ist auf direktem Wege mit jeder anderen verbunden.

Für dieses Problem gibt es ein sehr einfaches Verfahren zur Kostensenkung: Wir entfernen aus dem jeweiligen Graphen die teuerste Kante, die wir überhaupt entfernen können, ohne daß das Netz in zwei unzusammenhängende Teile zerfällt, und das immer wieder, bis nichts mehr geht. Für die 16 Städte des Odysseus ergibt sich nach diesem Verfahren ein Netzwerk mit einer Gesamt-

## Alle Wege führen (am Ende) nach Ithaka

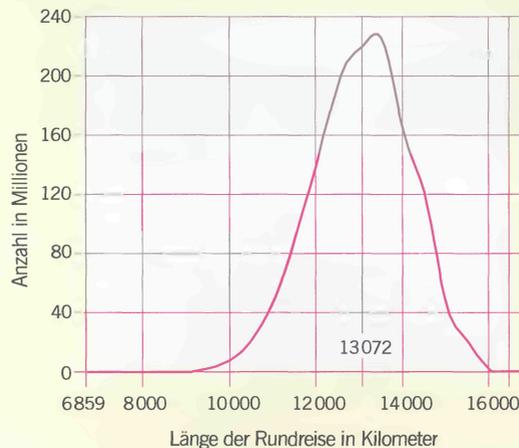
Wie viele Touren durch  $n$  Städte gibt es?

Die Anzahl aller möglichen Rundreisen durch eine gegebene Anzahl von Städten berechnet man mit einem Induktionsargument. Nehmen wir an, wir hätten bereits eine Liste aller Touren durch  $n-1$  Städte. Daraus konstruieren wir eine entsprechende Liste für  $n$  Städte, indem wir in eine  $(n-1)$ -Städte-Tour einen Abstecher zur  $n$ -ten Stadt einfügen. Dafür gibt es  $n-1$  Möglichkeiten: sämtliche Teilstrecken von einer Stadt zur nächsten. Also ist die Anzahl der  $n$ -Städte-Touren gleich  $(n-1)$  mal der Anzahl der  $(n-1)$ -Städte-Touren. (Bei diesem Konstruktionsverfahren kann eine Tour nicht auf zwei verschiedene Weisen entstehen, also gibt es auch keine Doppelzählungen.)

Für drei Städte gibt es genau eine Tour. (Ein Weg durch alle Städte und derselbe Weg in Gegenrichtung gelten als ein und dieselbe Tour; das genannte Konstruktionsverfahren stellt sicher, daß aus jedem Paar entgegengesetzt durchlaufener Touren stets nur genau eine in der Liste vertreten ist.) Für vier Städte gibt es drei Rundreisen, für fünf Städte  $3 \times 4 = 12$ , allgemein

für  $n$  Städte  $3 \times 4 \times \dots \times (n-1) = (n-1)!/2$  Touren. Für  $n=16$  ergibt diese Formel 653837184000 mögliche Routen.

Wir haben für das Problem des Odysseus die Längen all dieser Touren ausgerechnet. Die 10 kürzesten sind 6859, 6865, 6870, 6875, 6878, 6889, 6890, 6895, 6898 und 6900 Kilometer lang; die längste hat 16434 Kilometer. Die Graphik zeigt, wieviele Touren es zu einer gegebenen Länge gibt. Der Median ist 13072 Kilometer, das heißt, die Hälfte aller Touren ist länger, die andere Hälfte kürzer als dieser Wert. Die häufigste Tourlänge, 13359 Kilometer, ist mehr als 229 Millionen mal vertreten.



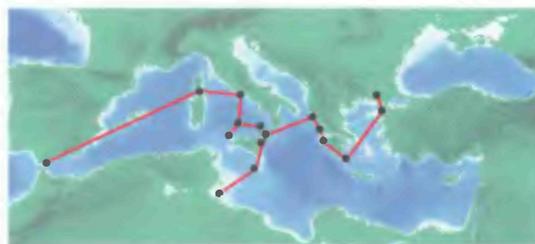
MARTIN GRÖTSCHIEL, MANFRED PADBERG

länge von 4540 Kilometern (siehe Karte unten).

Der beschriebene Algorithmus wählt also in jedem Moment unter den verfügbaren Möglichkeiten die günstigste, in diesem Falle die mit der größten Kostensenkung, ohne Rücksicht darauf, ob er sich damit nicht eine insgesamt gesehen viel bessere Lösung verbaut. Ein solches Verhalten – nehmen, was man kriegen kann, ohne auf die Spätfolgen zu sehen – nennt man gierig (*greedy*). Normalerweise ist es sinnvoll, auf der Suche nach dem größten Vorteil vorübergehend einen Nachteil in Kauf zu nehmen. Aber in diesem Fall ist blinde Gier genau das Richtige! Man kann beweisen, daß dieser Algorithmus stets eine optimale, das heißt kostenminimale Lösung des Problems liefert.

Wie viele mögliche Lösungen gibt es für dieses Problem? Bei  $n$  Städten ist jeder Graph mit genau  $n-1$  Kanten, der keinen geschlossenen Rundweg

durch einen Teil der Knoten enthält, eine mögliche Lösung. Solche Graphen heißen aufspannende Bäume (*spanning trees*) oder Gerüste (vergleiche Spektrum der Wissenschaft, April 1995, S. 10). Arthur Cayley (1821 – 1895) bewies bereits 1889, daß es für  $n$  Städte genau  $n^{n-2}$  verschiedene aufspannende Bäume gibt. Für 16 Städte beträgt diese Zahl 72057594037927936. Das ist beträchtlich mehr als die Anzahl der möglichen Rundreisen im Problem des Odysseus; und das gilt allgemein für beliebige (große) Städtezahlen. Trotzdem läßt sich un-



Das kostengünstigste Kommunikationsnetz zwischen den 16 Städten des Odysseus

ter diesen vielen Lösungen eine optimale auf einem PC innerhalb weniger Millisekunden errechnen.

Wenn es also nicht die Anzahl der möglichen Lösungen ist: Was macht das TSP zu einem so schwierigen Problem? Um ehrlich zu sein, wir kennen die Antwort selber nicht. Immerhin gibt es die Komplexitätstheorie, die zwischen „leichten“ Problemen wie dem des aufspannenden Baumes und „schwierigen“ wie dem TSP zu unterscheiden hilft. Die Theorie liefert ein unangenehmes Ergebnis: Sehr viele praktisch relevante kombinatorische Probleme sind als „schwierig“ zu klassifizieren.

Schon durch eine geringfügige Veränderung kann aus einem leichten Problem ein schwieriges werden. Nehmen wir an, wir legten bei unserem Kommunikationsnetzwerk Wert auf eine gewisse Ausfallsicherheit: Es soll auch dann noch funktionieren, wenn ein Feuer, ein Erdbeben oder auch nur der Fehlgriff eines Baggers eine Leitung zerstört. Ein aufspannender Baum bietet solche Sicherheit ja gerade nicht: Durch eine Unterbrechung an irgendeiner Stelle zerfällt das Netz in zwei Teile, die nicht mehr miteinander reden können.

Solch naheliegende Aspekte werden in der Realität nicht immer berücksichtigt. Als in den regionalen Telefonnetzen der USA die Kupferdrähte durch Glasfaserkabel ersetzt wurden, begnügte man sich wegen der hohen Kosten mit Minimallösungen – und hatte hinterher katastrophale Ausfälle zu beklagen, so den völligen Zusammenbruch des Chicagoer Telefonnetzes im Jahre 1988. Daraufhin waren sehr kostspielige Nachbesserungen vorzunehmen.

Je mehr zusätzliche Leitungen es gibt, desto größer ist die Ausfallsicherheit. Das Optimale unter diesem Aspekt ist der vollständige Graph; aber der ist zu teuer. Man muß also die Ausfallsicherheit gegenüber den Kosten der Konstruktion und Wartung abwägen. In aller Regel gibt es dabei wichtige und weniger wichtige Verbindungen beziehungsweise Städte; man wird Prioritäten vergeben. Solche Zusatzbedingungen sind mühelos in die mathematische Formulierung des Problems einzubringen; das ist das Schöne an den kombinatorischen Ansätzen. Leider wird dabei aus einem leichten Problem ein schwieriges – ebenso schwierig wie das TSP im allgemeinen Fall.

Eine mögliche (und übliche) Formulierung des Problems ist folgende. Wir vergeben Wichtigkeitswerte für die einzelnen Städte: 0 für die unbedeutenden, 1 für die wichtigeren, 2 für die großen Städte und

so weiter, und fordern, daß eine Stadt der Wichtigkeit  $j$  mit jeder Stadt gleicher oder größerer Wichtigkeit verbunden bleiben soll, selbst wenn  $j$  beliebige Direktverbindungen ausfallen. Unter allen Netzen, die diese Bedingung erfüllen, ist dasjenige mit den geringsten Kosten gesucht. In ähnlicher Form kann man Forderungen nach Sicherheit gegenüber dem Ausfall von Knoten oder von Knoten und Kanten zugleich in das Problem einbringen (Kasten Seite 36).

Probleme der Praxis enthalten häufig noch eine Fülle weiterer Nebenbedingungen; zum Beispiel kann eine Leitung oder ein Verbindungsrechner nur eine gewisse Anzahl von Gesprächen zugleich bewältigen. Durch sie wird die Anzahl der zulässigen Lösungen noch kleiner, aber das Problem nicht einfacher – im Gegenteil.

Soweit die Probleme. Wie findet man nun gute oder gar optimale Lösungen, und wie erfährt man, wie gut eine bestimmte Lösung ist, die der Computer ausspuckt? Bei allen – wichtigen – Unterschieden zwischen den Problemklassen gibt es einige Grundprinzipien.

### Obere und untere Schranken

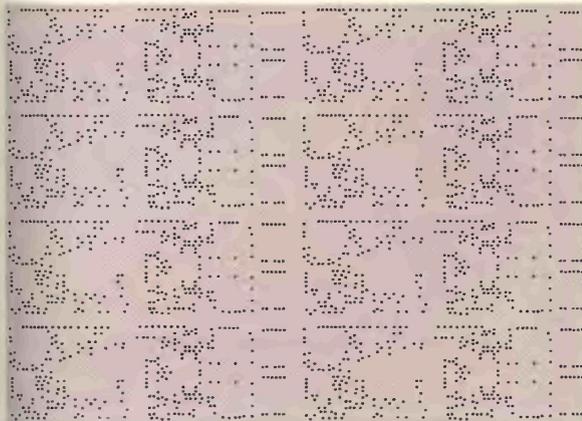
Gehen wir zunächst davon aus, daß eine gewisse Zielfunktion – Reiseweg, Kosten und so weiter – zu minimieren ist. (Maximierungsprobleme unterscheiden sich von Minimierungsproblemen nur durch das Vorzeichen.) Wir tragen in Gedanken den Wert jeder überhaupt zulässigen Lösung auf der reellen Zahlengeraden ein. Daß es eine optimale Lösung

geben muß, ist klar: Es gibt ja nur endlich viele Lösungen, und unter endlich vielen reellen Zahlen gibt es stets eine kleinste. Wir wissen nur nicht, wo auf der Zahlengeraden sie liegt. Nun gilt es diesen unbekanntem Wert irgendwie einzugrenzen, obere und untere Schranken auf der Zahlengeraden zu finden. Ohne die optimale Lösung zu kennen, wüßte man dann immerhin, daß sie nicht schlechter ist als die obere Schranke, aber auch nicht besser als die untere.

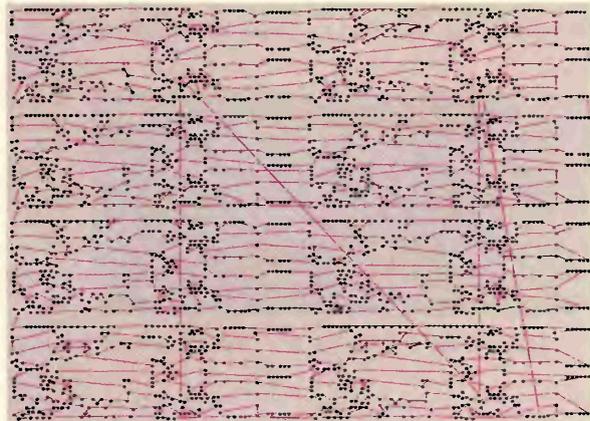
Obere Schranken zu finden ist nicht schwer. Jede zulässige Lösung ist eine, denn eine optimale Lösung ist *per definitionem* nicht schlechter als diese beliebige Lösung. Interessant sind natürlich Lösungen, die dem Optimum möglichst nahekommen. Algorithmen, die (möglichst gute) zulässige Lösungen und somit obere

## Die lange Rundreise des Bohrers

Mit der richtigen Reihenfolge spart man fast die Hälfte der Zeit



Die Aufgabe ...



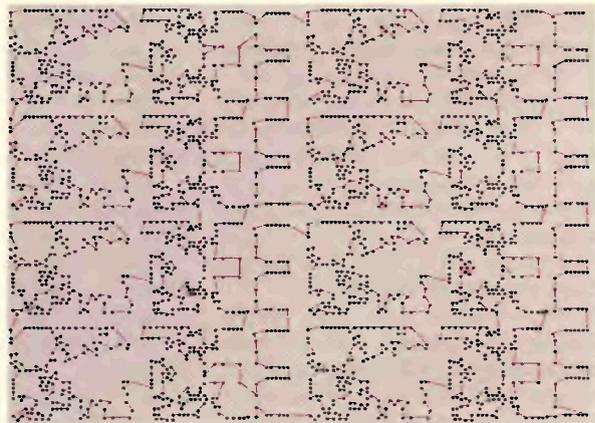
... irgendeine Lösung ...

Es sind mit einem mechanischen oder einem Laserbohrer 2392 Löcher in eine rechteckige Leiterplatte zu bohren. Durch sie sollen im nächsten Produktionsschritt die Anschlüsse der elektrischen Bauteile geführt werden. Derartige Leiterplatten finden sich in Waschmaschinen, Transistorradios, Fernsehgeräten, Computern und zahlreichen anderen Geräten. Genau wie Odysseus muß der Bohrer jedes Loch (jede Stadt) genau einmal anfahren. Die Bohrzeit pro Loch ist konstant; an der Gesamtbohrzeit kann man nur sparen, indem man die Gesamtentfernung, die der Bohrer zurücklegen muß, minimiert.

Wieder ist es, wie beim Problem des Odysseus, einfach, eine beliebige Rundreise für den Bohrer zu finden. Ein Planer in der Industrie hat die rechts oben abgebildete Lösung entworfen, die offensichtlich noch Verbesserungsreserven enthält.

Bei der rechts unten dargestellten optimalen Lösung beträgt die zurückzulegende Gesamtentfernung 378032 statt 718876 (in willkürlichen Einheiten) – eine Ersparnis von fast 50 Prozent! Eine Workstation benötigt dafür weniger als eine Stunde. Dabei wurde angenommen, daß die Fahrzeit des Bohrers zwischen zwei Bohrungen proportional zu ihrer Entfernung ist. Andere Annahmen sind ebenfalls sinnvoll. Wenn, wie häufig, der Bohrer von zwei unabhängigen Motoren in  $x$ - und in

$y$ -Richtung bewegt wird, ist das Maximum der Entfernungen in  $x$ - und in  $y$ -Richtung das realistischere Maß für die Fahrzeit.



... und die optimale Lösung

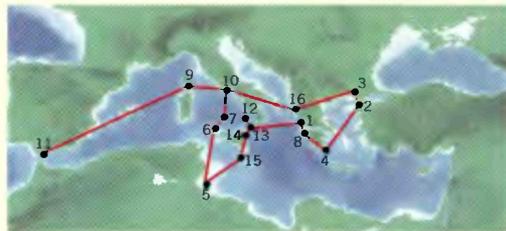
MARTIN GRÖTSCHEL, MANFRED PAUBERG

## Ausfallsichere Kommunikationsnetze

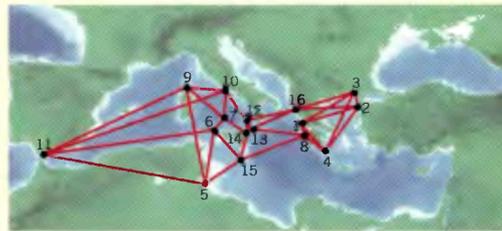
Meistens erfüllt das optimale Netz nicht nur die geforderten Bedingungen, sondern noch einige mehr

Nehmen wir an, wir hätten die 16 Mittelmeerstädte des Odysseus durch ein ausfallsicheres Netzwerk miteinander zu verbinden. Aus historischen Gründen erklären wir Ithaka (1), Troja (2), Djerba (5), Circeo (10) und Messina (13) zu bedeutenden Städten und teilen ihnen die Wichtigkeit 1 zu, während alle anderen Städte den Wert 0 erhalten. Es sollen also alle bedeutenden Städte miteinander verbunden bleiben, selbst wenn eine beliebige Direktverbindung ausfällt.

Wenn man davon ausgeht, daß die Konstruktionskosten für eine direkte Verbindung proportional zur Länge der Verbindung sind, erhalten wir als Optimallösung ein Netzwerk mit einer Länge von 5842 Kilometern (unten). Dabei genießen nicht nur die fünf auserwählten Städte, sondern alle bis auf 9, 11 und 12 die erhöhte Ausfallsicherheit. Heutzutage versuchen Netzwerkplaner ein vernünftiges Gleichgewicht zwischen Kosten und Sicherheit durch Probieren zu finden, in-



Gefordert: Rang 1 für die Städte 1, 2, 5, 10 und 13



Gefordert: Rang 3 für alle Städte

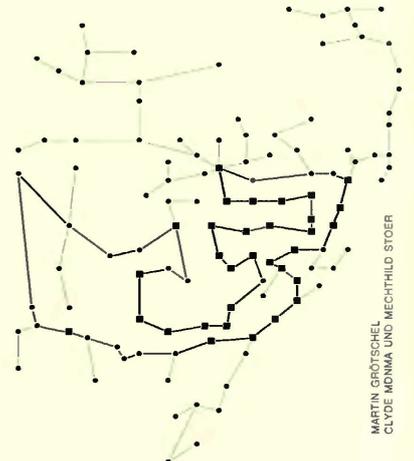
dem sie die Parameter, etwa die Wichtigkeitswerte, variieren.

Wenn wir mehr für die Sicherheit auszugeben bereit sind, beispielsweise jedem Knoten unseres Problems die Wichtigkeit 3 zuordnen, erhalten wir das oben abgebildete kostenminimale Netzwerk mit einer Länge von 16112 Kilometern. Es würde selbst beim gleichzeitigen Ausfall von drei direkten Kommunikationsverbindungen funktionsfähig bleiben. Auch beim Ausfall eines der Knotenpunkte im Netzwerk (zum Beispiel durch Feuer in der Vermittlungsstelle) bleiben die anderen Knoten über einen Kommunikationsweg miteinander verbunden. Erst wenn die Knoten 8 (Zakynthos) und 16 (Korfu) gleichzeitig ausfallen, ist das Netz unterbrochen.

In einem realistischen Fall, nämlich einem regionalen Telefonnetz für den östlichen Teil der USA, sieht die kostenminimale Lösung so aus wie rechts abgebildet.

Knoten der Wichtigkeit 0 sind durch Kreise, solche der Wichtigkeit 1 durch Quadrate gekennzeichnet. Dieses Kommunikationsnetzwerk umfaßt 116 Knoten. In diesem Fall sind die Kosten einer Verbindung nicht proportional der Streckenlänge; Berge oder Flüsse auf dem Weg können die Kosten erheblich in die Höhe treiben.

Die Beispiele dieses Kastens wurden mit einem Code berechnet, den Martin Grötschel, Clyde Monma von der Forschungsabteilung von Bell Communications in Morristown (New Jersey) und Mechthild Stoer von der Forschungsabteilung der norwegischen Telefongesellschaft in Kjeller entwickelt haben.



MARTIN GRÖTSCHEL  
CLYDE MONMA UND MECHTHILD STOER

Telephoneäst für den Osten der USA

re Schranken für den Optimalwert liefern, heißen im Fachjargon Heuristiken.

Auch die Auskunft, daß keine Lösung besser sein kann als ein gewisser Wert, eben eine untere Schranke, ist hilfreich: Treffen die obere und die untere Schranke zusammen, ist damit bewiesen, daß die zur oberen Schranke gehörige Lösung optimal ist. Wenn das – wie meistens – nicht der Fall ist, weiß man immerhin, wie weit man höchstens noch vom Optimum entfernt ist. Wenn diese Abweichung hinreichend klein ist, liefert das einen guten Grund, die Arbeit einzustellen: Selbst im Erfolgsfall würde die weitere Suche nicht viel einbringen.

Für die Suche nach einer unteren Schranke ist ebenfalls eine geeignete mathematische Formulierung zu finden. Besonders geschickt ist es, abwechselnd nach oberen und nach unteren Schranken zu suchen und bei jedem Suchschritt die Informationen aus den vorangegangenen

Schritten mitzuverwerten. Dieser Kombinationsstrategie sind die enormen Fortschritte der letzten Jahre zu verdanken.

Techniken zur Ermittlung von beiderlei Schranken wollen wir nun wieder am Beispiel des Odysseus diskutieren. Für andere schwierige Probleme hört sich die Geschichte nur geringfügig anders an; allerdings kann durch diese kleinen technischen Details das Problem abermals erheblich schwerer werden.

Für die Suche nach einer brauchbaren oberen Schranke ist zunächst abermals Gier – genauer: kurzichtiges Streben nach Kostenminimierung – ein guter Ratgeber. Von jeder Stadt aus fahre man in die nächstgelegene noch nicht besuchte Stadt (Nearest-Neighbour-Heuristik). Wenn keine Stadt mehr übrig ist, kehre man zum Ausgangspunkt zurück.

Wer dieses Verfahren mit Papier und Bleistift an beliebigen Ansammlungen von Städten durchspielt, bemerkt sehr

schnell zweierlei. Erstens: Das Resultat einer Heuristik hängt stark vom Ausgangspunkt ab. Zweitens: Eine solche Lösung ist leicht zu verbessern. Wenn sich etwa zwei Kanten überschneiden, ist es günstiger, in dem Viereck aus den vier beteiligten Städten nicht die Diagonalen entlang zu reisen, sondern zwei gegenüberliegende Seiten. Möglicherweise gewinnt man auch etwas, wenn man zwei solche Seiten eines Vierecks durch die beiden anderen ersetzt. Es gibt weitere Verbesserungsschritte dieser Art.

Um aus solchen Ideen eine brauchbare Methode zu machen, muß man etwas gegen die ungünstigen Folgen der Gier tun. Wer immer nur darauf besteht, eine Lösung zu verbessern, läuft Gefahr, in einem schlechten „lokalen Optimum“ steckenzubleiben: einer Lösung, die nicht besonders gut ist, aber durch keine einzelne der verfügbaren Änderungen besser wird. Man muß also – vorüberge-

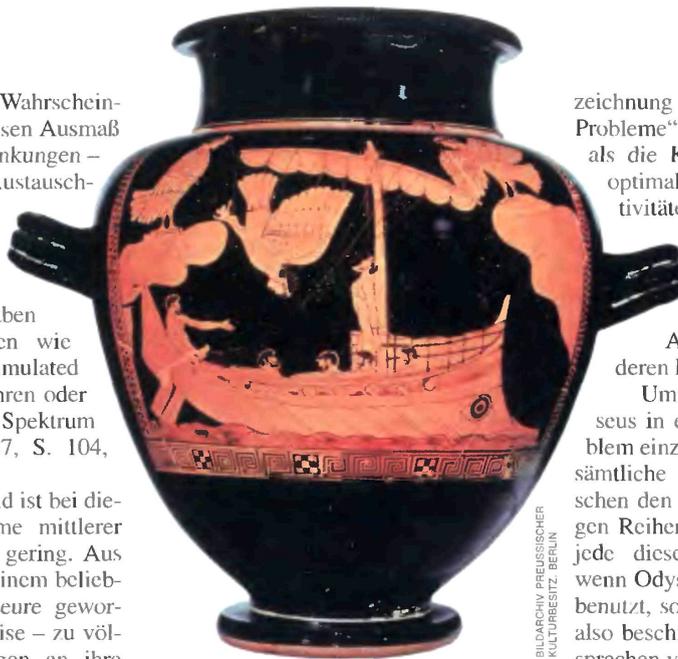
hend, mit einer gewissen Wahrscheinlichkeit, bis zu einem gewissen Ausmaß oder unter anderen Einschränkungen – zulassen, daß ein solcher Austauschschritt eine Lösung verschlechtert, oder ab und zu per Zufall eine völlig neue heuristische Lösung erzeugen. Derartige Methoden haben meist unterhaltsame Namen wie Monte-Carlo-Methoden, Simulated Annealing, Evolutionsverfahren oder Sintflut-Algorithmus (siehe Spektrum der Wissenschaft, Juli 1987, S. 104, und in diesem Heft S. 22).

Der Programmieraufwand ist bei diesen Methoden für Probleme mittlerer Größe in der Regel relativ gering. Aus diesem Grunde sind sie zu einem beliebten Tummelplatz für Amateure geworden, was – unglücklicherweise – zu völlig überzogenen Erwartungen an ihre Güte und Durchführbarkeit geführt hat. Für TSPs mit Zehntausenden oder Hunderttausenden von Knoten (solche Größenordnungen sind nicht ungewöhnlich) müssen dann doch komplizierte Datenstrukturen und raffinierte Tricks der Informatik hinzukommen, damit man brauchbare Lösungen in akzeptabler Zeit findet. Andererseits sind heuristische Algorithmen, die in einem längeren Experimentierprozeß dem speziellen Problemtyp angepaßt wurden, aufgrund gegenwärtiger technologischer Grenzen zum Arbeitspferd für die Lösung praktischer Probleme geworden und auch für die exakten Optimierungsmethoden in mancherlei Hinsicht hilfreich.

Wie findet man nun eine untere Schranke zum Optimalwert? Die – nur scheinbar paradoxe – Grundidee ist, das Problem größer zu machen, damit es einfacher wird. Das bedeutet: Zu unserem Problem konstruieren wir ein weiteres, das mehr Lösungen hat: Jede Lösung des alten Problems ist auch eine des neuen (aber nicht notwendig umgekehrt). Zum Beispiel könnte man Nebenbedingungen des ursprünglichen Problems lockern oder außer Kraft setzen. Die Lösungsmenge des ursprünglichen Problems ist also in der des neuen Problems enthalten („eingebettet“); die Fachleute sprechen von *embedding techniques*.

Eine minimale Lösung des großen Problems hat *per definitionem* geringere Kosten als jede andere Lösung, also auch geringere als – oder höchstens gleich große wie – jede Lösung des ursprünglichen, eingebetteten Problems. Also ist sie das, was wir suchen: eine untere Schranke für den Optimalwert des ursprünglichen Problems.

Damit diese Idee etwas einbringt, sollte das einbettende Problem einerseits



BILDARCHIV PREUSSISCHER KULTURERBISITZ, BERLIN

**Wer unter mehreren Alternativen stets die im Moment verlockendste wählt, ist nicht fähig, um des großen langfristigen Vorteils willen auf den unmittelbaren zu verzichten: Das ist der Nachteil eines gierigen Algorithmus. Um diesen Fehler zu vermeiden und den Gesängen der Sirenen nicht anheimzufallen, läßt sich Odysseus von seinen Gefährten am Mast festbinden (Malerei auf einem attischen Krug um 450 vor Christus).**

einfach sein, damit seine minimale Lösung leicht zu finden ist, andererseits dem ursprünglichen möglichst ähnlich. Nur dann kann man hoffen, daß auch die Minimallösung des großen Problems derjenigen des eingebetteten Problems nahe kommt. Es ist keine Kunst, ein TSP in irgendein größeres Problem einzubetten. Man könnte Odysseus die Freiheit geben, sich überhaupt nicht zu bewegen. Nur ist die untere Schranke, die man daraus gewinnt – jede Tour ist länger als null Kilometer –, nicht besonders aufschlußreich.

#### Untere Schranken aus der linearen Programmierung

Unter den verschiedenen Einbettungstechniken wollen wir hier lediglich die erfolgreichste diskutieren. Sie basiert auf der linearen Programmierung, einer der bedeutsamsten Techniken, welche die angewandte Mathematik seit dem Zweiten Weltkrieg hervorgebracht hat (siehe „Wirtschaftsfaktor lineare Programmierung“ von Robert G. Bland, Spektrum der Wissenschaft, August 1981, S. 118).

Der – überkommene – Name ist irreführend: es geht nicht in erster Linie ums Programmieren. Eine zutreffende Be-

zeichnung wäre „Optimierung linearer Probleme“. Häufig wird sie interpretiert als die Kunst, knappe Ressourcen in optimaler Weise wirtschaftlichen Aktivitäten zuzuordnen. Sie ist heute Standardwerkzeug zur Lösung von Planungsproblemen bei Fluggesellschaften, Ölraffinerien, Banken, Autoherstellern und vielen anderen Firmen.

Um nun das Problem des Odysseus in ein lineares Optimierungsproblem einzubetten, notieren wir zunächst sämtliche 120 Direktverbindungen zwischen den 16 Städten – in einer beliebigen Reihenfolge – und schreiben neben jede dieser Verbindungen eine Eins, wenn Odysseus sie auf seiner Rundreise benutzt, sonst eine Null. Jede Tour wird also beschrieben durch 120 Zahlen. Wir sprechen von einem Vektor der Dimension 120, denn wir wollen nun jede Folge von 120 Zahlen als einen Punkt (oder eben einen Vektor) in einem Raum der Dimension 120 auffassen, ebenso wie jede Folge dreier Zahlen ( $x, y, z$ ) einen Punkt im gewöhnlichen dreidimensionalen Raum beschreibt.

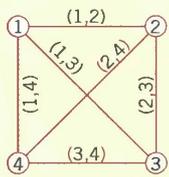
Jede Tour entspricht also einem Punkt in diesem 120-dimensionalen Raum – aber nicht umgekehrt! Damit ein Punkt einer Tour entspricht, müssen genau 16 der Komponenten gleich eins und die übrigen 104 gleich null sein. Weitere Bedingungen sind zu erfüllen (siehe Kasten Seite 38).

Wir erinnern uns, daß es genau 653 837 184 000 mögliche Rundreisen für Odysseus gibt. Aus so vielen Punkten im 120-dimensionalen Raum möchten wir einen auswählen, bei dem die Summe der Kilometerzahlen, die zu den Komponenten mit dem Wert 1 gehören, möglichst gering ist. Wir multiplizieren also jede Komponente unseres Vektors – 1 oder 0 – mit der zugehörigen Kilometerzahl, addieren alle Produkte auf und erhalten die Gesamtlänge der Tour.

Jetzt kommt der entscheidende Schritt. Wir fassen die Komponenten unseres Vektors als Variable auf, die außer 0 und 1 auch jeden beliebigen Wert dazwischen annehmen dürfen. Das ist vom ursprünglichen Problem her gesehen ziemlich albern: Was soll das heißen, daß Odysseus ein halbes Mal von Ithaka nach Troja fliegt oder 0,17 mal von Messina nach Gibraltar? Es ist aber für unsere Zwecke sinnvoll, denn wir erweitern zwar gewaltig die Menge der zulässigen Lösungen, bringen aber zugleich das Problem in die Reichweite der linearen Programmierung. Die Zielfunktion – die Gesamt-Tourlänge – ist nämlich eine lineare Funktion der 120 Variablen: Jede

## Die Rundreise als lineares Optimierungsproblem

Man erweitert die Aufgabe, damit sie hinterher einfacher ist



Betrachten wir zunächst ein einfaches Problem mit vier Städten, die mit 1 bis 4 nummeriert sind. Der vollständige Graph zu vier Knoten enthält sechs Kanten. Jede Kante trägt die Nummern der Städte, die sie verbindet (die kleinere Nummer zuerst). Man kann eine Rundreise eindeutig kennzeichnen, indem man den Kanten, die zur Rundreise gehören, den Wert 1 zuweist, den anderen den Wert 0:

Kante	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
	(1, 0, 1, 1, 0, 1)					
	(1, 1, 0, 0, 1, 1)					
	(0, 1, 1, 1, 1, 0)					

Jede Tour wird so zu einem Vektor im 6-dimensionalen Raum, der genau vier Einsen und sonst Nullen enthält. Entsprechend wird jede Tour durch die 16 Städte des Odysseus zu einem Vektor aus 120 Zahlen, von denen genau 16 gleich 1 und die übrigen gleich 0 sind. Ein Vektor, der nur aus Einsen und Nullen besteht, heißt Inzidenzvektor.

Nicht jeder Inzidenzvektor mit 16 Einsen und sonst Nullen entspricht einer Rundreise. Von den Kanten, die zu einer Stadt gehören, dürfen genau zwei mit einer Eins belegt sein: der Weg, auf dem Odysseus die Stadt betritt, und der, auf dem er sie wieder verläßt. Das läßt sich als Nebenbedingung in Form einer linearen Gleichung formulieren: Die Summe aller Komponenten des Vektors, die den zu einer Stadt gehörigen Kanten entsprechen, muß gleich 2 sein.

Ein Inzidenzvektor, der auch diese Bedingung erfüllt, muß jedoch immer noch nicht einer Rundreise entsprechen. Es könnte sein, daß sich die ausgewählten Kanten zu mehreren kleinen Touren statt zu einer großen zusammensetzen (oben). Man muß



Lösung des erweiterten Problems – aber nicht des ursprünglichen

also weitere Bedingungen stellen, um die „richtigen“ von den „falschen“ Inzidenzvektoren zu unterscheiden.

Man nehme etwa die sieben westlichen Städte, die im Bild durch eine Teiltour verbunden sind. Unter allen Kanten, die diese Städte untereinander verbinden, sind im Inzidenzvektor sieben mit einer Eins belegt. Das ist mindestens eine zuviel; sieben Kanten zwischen sieben Städten, wobei jede Stadt nur von zwei Kanten berührt wird, müssen zwangsläufig eine geschlossene Rundreise (oder gar mehrere) durch diese Städte enthalten, was wiederum eine komplette Rundreise durch alle 16 Städte ausschließt. Allgemein läßt sich diese Bedingung so formulieren: Zu jeder Teilmenge von  $k$  Städten darf der vollständige Graph dieser  $k$  Städte höchstens  $k-1$  Kanten enthalten, die mit einer Eins belegt sind. Das ist eine lineare Ungleichung: Die Summe aller Komponenten des Inzidenzvektors, die zu den Kanten des vollständigen Graphen einer Teilmenge aus  $k$  Städten gehören, muß kleiner oder gleich  $k-1$  sein.

Jede Teilmenge von (mindestens drei) Städten liefert eine solche Teiltour-Ausschlußbedingung; das sind bei  $n$  Städten größenordnungsmäßig  $2^{n-1}$  Stück (nicht  $2^n$ , denn die Ausschlußbedingung für eine Teilmenge und die für ihr Komplement sind im wesentlichen dasselbe). Für sehr große Städteanzahlen sind das weitaus mehr, als man explizit berücksichtigen kann.

Änderung in einer der Variablen wirkt sich mit einem Proportionalitätsfaktor – nämlich der Länge der entsprechenden Teilstrecke – auf den Wert der Zielfunktion aus. Die Nebenbedingungen sind ebenfalls linear (siehe Kasten oben).

Für die Suche nach dem Minimum einer linearen Funktion mit linearen (Gleichungs- oder Ungleichungs-)Nebenbedingungen gibt es ein effizientes Verfahren, den Simplex-Algorithmus. (Die seit etwa zehn Jahren intensiv erforschten Innere-Punkte-Verfahren bringen für unsere Zwecke noch keine Vorteile ein.)

Ein wenig Geometrie ist hilfreich, auch wenn unser Vorstellungsvermögen nur drei statt 120 Dimensionen erfaßt. Nehmen wir alle Punkte (Vektoren) mit Koordinaten zwischen 0 und 1. Sie bilden einen Würfel der Kantenlänge 1 mit einer Ecke im Nullpunkt. Die Ecken dieses Einheitswürfels sind die Vektoren, deren Komponenten sämtlich gleich 0 oder 1 sind. Das gilt auch in 120 Dimensionen, nur hat der Einheitswürfel jetzt

$2^{120}$  Ecken; die kann man unmöglich eine nach der anderen absuchen. Eine lineare Ungleichungs-Nebenbedingung ist wie ein Messer, das von dem Würfel (denken wir ihn uns aus Käse) ein Stück abschneidet. Es entsteht ein von ebenen Flächen begrenzter Körper.

Im 120-dimensionalen Raum ist die Menge der zulässigen Punkte, das heißt derjenigen, die sämtliche (linearen) Nebenbedingungen erfüllen, die Verallgemeinerung eines solchen Körpers: ein (konvexes) Polytop. Das Minimum einer linearen Zielfunktion auf einem Polytop wird stets in einer Ecke angenommen (oder allenfalls in mehreren Ecken und allem, was dazwischen ist, zugleich). Der Simplex-Algorithmus sucht systematisch (auch hier einem Gier-Prinzip folgend) die Ecken des Polytops ab, bis er nicht mehr weiterkommt; die Ecke, an der er stehenbleibt, ist dann das gesuchte Minimum (oder eines von mehreren gleichberechtigten).

Leider handelt es sich nur um das Minimum des erweiterten Problems,

nicht um das des ursprünglichen. Im allgemeinen hat also der Lösungsvektor, den der Simplex-Algorithmus liefert, keine ganzzahligen Koordinaten. Man muß demnach das Käsemesser nochmals ansetzen, um aus dem Polytop des linearen Problems das „echte“ Polytop des ursprünglichen Problems zu machen; das ist das kleinste Polytop, das sämtliche Inzidenzvektoren von Touren enthält (Kasten Seite 40).

Das gelingt jedoch nicht mit einem Schnitt. Schlimmer noch: Wir wissen nicht einmal genau, wie wir schneiden müßten. Explizite lineare Programme für Rundreiseprobleme sind bislang für höchstens 9 Städte bekannt. Genauer gesagt: Das mit dem 9-Städte-TSP assoziierte Polytop hat 9 Gleichungen und 42 104 442 Ungleichungen mit 36 Variablen, entsprechend den 36 möglichen direkten Verbindungen zwischen 9 Punkten. Für Polytope mit 8 (7 und 6) Städten gibt es entsprechend 194 187 (3437 beziehungsweise 100) Ungleichungen mit 28 (21 beziehungsweise 15) Variablen.

Für zehn Städte sind es wahrscheinlich 51043900866 Ungleichungen, mit Sicherheit nicht weniger; der Beweis für diese Vermutung steht noch aus. Für größere Rundreiseprobleme ist noch nicht einmal theoretisch ein vollständiges Sortiment von Ungleichungen bekannt, welches das echte Polytop beschreibt. Und selbst wenn es bekannt wäre, würde es, da zu groß, nicht viel helfen. Für allgemeine, größere Städtezahlen wird es möglicherweise nie gefunden werden, und das wäre ein weiterer Beweis der enormen Schwierigkeit des *travelling salesman problem*.

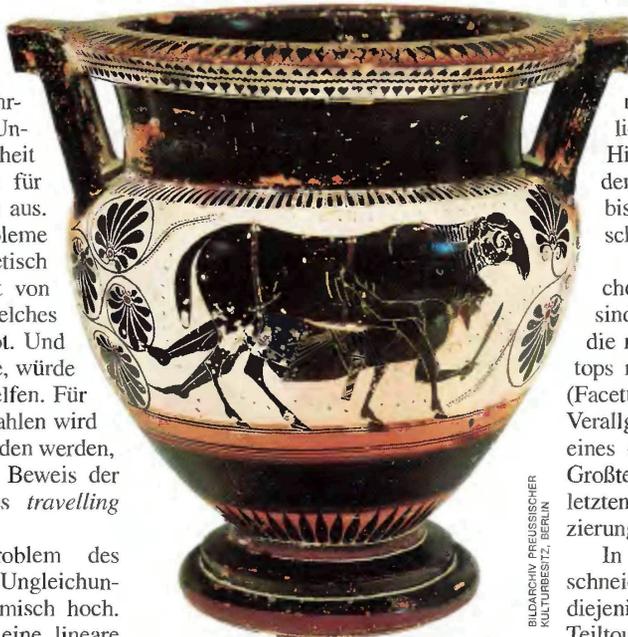
Für das 16-Städte-Problem des Odysseus ist die Anzahl der Ungleichungen auf jeden Fall astronomisch hoch. Immerhin kann man stets eine lineare Ungleichung (entsprechend der Verallgemeinerung eines ebenen Schnittes) finden, die den bisher ermittelten Minimalpunkt vom echten Polytop trennt. Diese Ungleichung fügt man dem System der bisherigen Ungleichungen hinzu, optimiert aufs neue und findet einen neuen Minimalpunkt, der dem echten Polytop näher liegt.

Wenn man Glück hat, liegt er sogar auf dem echten Polytop; dann hat man eine minimale Lösung des ursprünglichen Problems erzielt. In der Regel aber hat er wieder keine ganzzahligen Koordinaten, und man muß das Verfahren wiederholen.

Ein weiteres Problem kommt hinzu. Man kann im allgemeinen noch nicht einmal mit dem Polytop der zulässigen Punkte des linearen Optimierungsproblems anfangen. Dazu müßte man nämlich sämtliche Nebenbedingungen berücksichtigen, und das sind für große Städtezahlen ebenfalls astronomisch viele. Gleichwohl gelingt es, weitaus größere Probleme mit Methoden der linearen Programmierung optimal zu lösen. Wie ist das möglich?

Wir ersetzen abermals das Problem durch ein größeres, einfacheres: Von den astronomisch vielen Nebenbedingungen verwenden wir nur eine sehr kleine Teilmenge (wir beschneiden den Käsewürfel nur sehr grob), finden einen Lösungsvektor, das heißt eine optimale Ecke des zu großen Polytops, und bessern dann erst nach; Wir führen eine Nebenbedingung ein, die mit der bisher gefundenen Ecke gleich ein möglichst großes Stück Käse abschneidet.

Beim Problem des Odysseus können wir uns zunächst auf die (Gleichungs-) Nebenbedingungen beschränken, daß jede Stadt nur mit zwei Wegen an der Tour beteiligt sein darf: dem Weg in die Stadt hinein und dem aus ihr heraus.



BILDARCHIV PREUSSISCHER KULTURERBESITZ, BERLIN

**Polyphem und das Polytop: Der Suchalgorithmus befindet sich in der Situation eines Blinden. Er hat keinen allgemeinen Überblick, sondern kann sich nur vorantasten, und immer nur in eine Richtung auf einmal. Aus Zeitgründen überprüft er nicht jedesmal, ob jede Nebenbedingung eingehalten wird, und muß deshalb Verstöße hinnehmen. So entwischt Odysseus, unter den Bauch des Widders gekrallt, dem gefräßigen Zyklopen – aus dessen Sicht eine unzulässige Lösung (Darstellung des sogenannten Sappho-Malers auf einem Gefäß um 510 vor Christus).**

(Eine Gleichungs-Nebenbedingung reduziert im Gegensatz zu einer Ungleichung die Dimension des Polytops, was sich leider mit Käse nicht mehr veranschaulichen läßt.)

Die Lösung des entsprechenden linearen Programms liefert einen Minimalwert von 6113 Kilometern für die Zielfunktion. Das ist immerhin schon eine brauchbare untere Schranke. Allerdings besteht diese Lösung (Kasten Seite 38) aus mehreren Teiltouren. Da sie noch kei-

ne zulässige Lösung des ursprünglichen Problems ist, muß man durch Hinzunahme einer Nebenbedingung den entsprechenden Punkt von dem bisher gewonnenen Polytop abschneiden.

Es steht eine große Anzahl solcher Schnitte zur Auswahl; darunter sind „bestmögliche“, das heißt solche, die mit den Facetten des echten Polytops möglichst viel gemeinsam haben. (Facetten sind die vieldimensionalen Verallgemeinerungen der Grenzflächen eines dreidimensionalen Polytops.) Ein Großteil der theoretischen Arbeit der letzten 20 Jahre hat sich auf die Identifizierung solcher Schnitte konzentriert.

In unserem Fall fügen wir als abschneidende Nebenbedingungen gerade diejenigen hinzu, welche die vier Teiltouren der bisher gefundenen Lösung ausschließen. Nun ist das um diese Ungleichungen erweiterte lineare Programm zu lösen; moderne Softwarepakete erledigen das schnell und effizient, weil sie die in der bisherigen Lösung enthaltene Information geschickt ausnutzen.

Es ergibt sich ein Lösungsvektor mit dem Zielfunktionswert 6228 Kilometer, der noch immer nicht eine passende Komplett-Rundreise für Odysseus ist (Kasten Seite 40, f). Die algorithmische Idee ist jetzt jedoch klar: Weiter so. Wir müssen nur noch iterieren. Wir aktivieren eine weitere kleine Teilmenge der astronomisch vielen Ungleichungen, die wir bislang ignoriert haben. Dadurch und durch die Lösung eines dritten linearen Programms erhalten wir eine Zielfunktion mit dem Optimalwert 6813,5 Kilometer und einen Lösungsvektor, der wiederum keine Rundreise für Odysseus ist (Kasten Seite 40, g). Er enthält nämlich für einige der 120 Variablen den Wert 1/2 – was für das erweiterte Problem ja zulässig ist. Eine weitere Iteration liefert

#### Literaturhinweise

*Ulysses Found.* Von E. Bradford. Harcourt, Brace & World, New York 1963.  
*The traveling salesman problem. A guided tour of combinatorial optimization.* E. L. Lawler et al. (Hrsg.). Wiley, Chichester 1985.  
*Design of Survivable Networks.* Von Mechthild Stoer. Lecture Notes in Mathematics, Bd. 1531. Springer, Berlin 1992.  
*Linear Optimization and Extensions.* Von Manfred Padberg. Springer, Berlin 1995.  
*The travelling salesman problem.* Von Michael Jünger, Gerhard Reinelt und

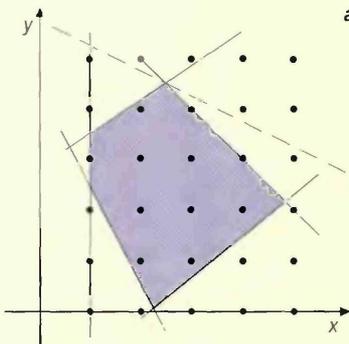
Giovanni Rinaldi in: *Annotated bibliographies in combinatorial optimization.* Mauro Dell'Amico et al. (Hrsg.). Wiley, Chichester, S. 199 – 221 (1997).

*On the solution of traveling salesman problems.* Von David Applegate et al. in: *Documenta Mathematica, Extra Volume, Proceedings of the ICM 98,* S. 645–656 (1998); <http://www.mathematik.uni-bielefeld.de/documenta/xvol-icm/ICM.html>.

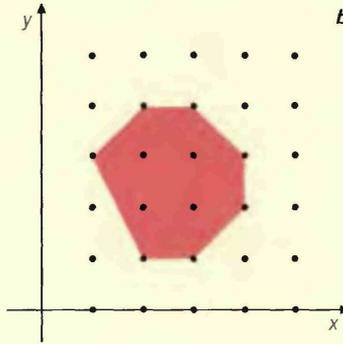
*Well-solvable special cases of the traveling salesman problem: A survey.* Von Rainer E. Burkard et al. in: *SIAM Review,* Bd. 40, S. 496–546 (1998).

## Lineare und ganzzahlige Optimierung

Polytope zurechtschneiden ist schwer – man weiß gar nicht, wo man anfangen soll



MARTIN GRÖTSCHHEL, MANFRED PADBERG



Der Anschaulichkeit zuliebe zeigen wir hier Beispiele in zwei (statt drei oder gar 120) Dimensionen. Es gibt also nur zwei Größen –  $x$  und  $y$  –, die man überhaupt variieren kann. Eine lineare (Ungleichungs-)Nebenbedingung entspricht einer Geraden in der  $x$ - $y$ -Ebene mit der Eigenschaft, daß alle Punkte, die auf einer Seite der Geraden liegen, unzulässig sind. Mehrere Nebenbedingungen grenzen so in der  $x$ - $y$ -Ebene ein Vieleck (im allgemeinen Fall: ein Polytop) ein, das aus allen zulässigen Lösungen des Problems besteht (a).

Die Kostenfunktion ist ebenfalls linear in den Variablen des Problems. Deswegen gibt es Geraden (im allgemeinen Fall: Hy-

perebenen), auf denen die Kostenfunktion konstant ist. Die gestrichelte Linie im Bild a zeigt beispielhaft eine solche Gerade; die Punkte links unterhalb von ihr sind alle teurer, die Punkte rechts oberhalb alle billiger als die Punkte auf der Geraden selbst. Die optimale Lösung des Problems – der billigste Punkt des Vielecks – muß eine Ecke des Vielecks sein: der einzige Punkt des Vielecks, der diese günstigste erreichbare Gerade konstanter Kosten trifft. Allenfalls könnte eine Vielecksseite parallel zu den Geraden konstanter Kosten sein; dann sind die beiden Endpunkte dieser Seite – und jeder Punkt dazwischen – gleichberechtigte optimale Lösungen. In jedem Fall genügt es, die Ecken des Vielecks (beziehungsweise Polytops) nach dem Optimum abzusuchen.

Nun ist das lineare Problem in unserem Falle nur die Erweiterung eines Problems, dessen zulässige Lösungen Vektoren mit ganzzahligen Koordinaten sind (schwarze Punkte). Im Problem des Odysseus sind die zulässigen ganzzahligen Werte sogar nur 0 und 1. Wir suchen also gar nicht die optimale Lösung des linearen Problems – die liegt nämlich im allgemeinen nicht auf einem solchen Gitterpunkt – sondern unter den Gitterpunkten innerhalb des Vielecks (Polytops) der zulässigen Lösungen den optimalen. Man müßte also dieses Vieleck so zurechtstutzen, daß es nur noch diese zulässigen Gitterpunkte enthält samt allem, was dazwischen ist: die konvexe Hülle der zulässigen Gitterpunkte (b).

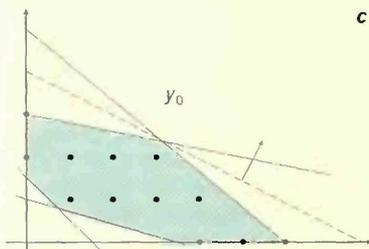
Zurechtstutzen bedeutet zusätzliche Nebenbedingungen einführen. Leider ist es extrem schwierig, sie alle zu finden, schon deshalb, weil es astronomisch viele werden können. Man geht deswegen iterativ vor:

Zu einem „falschen“, das heißt nicht-ganzzahligen Optimum  $y_0$  des linearen Problems (c) finde man eine Gerade (eine lineare Ungleichungs-Nebenbedingung), die das falsche Optimum ausschließt, aber alle zulässigen Gitterpunkte zulässig läßt. Unter den vielen Geraden mit dieser Eigenschaft gibt es „beste“, das heißt sol-

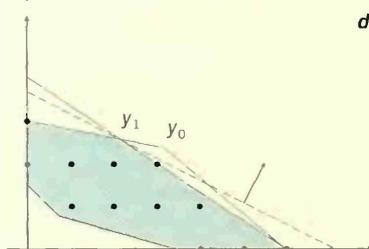
che, die von dem Vieleck ein maximal großes Stück abschneiden. Man löse das lineare Problem mit dieser zusätzlichen Nebenbedingung (d), finde ein Optimum  $y_1$ , das sich abermals als falsches Optimum herausstellt, und wiederhole den Prozeß solange, bis das Optimum auf einen Gitterpunkt zu liegen kommt (e). Das ist der Schnittebenen-Algorithmus.

Was in der Ebene einfach ist, nämlich eine beste Schnittgerade (entsprechend einer Schnittebene) zu finden, ist in hohen Dimensionen schwierig bis praktisch unmöglich. Im Jahre 1958 hat Ralph Gomory, damals an der Universität Princeton, später langjähriger Forschungschef von IBM und heute Präsident der Alfred-P.-Sloan-Stiftung, ein Verfahren angegeben, das garantiert nach endlich vielen Schritten zum Erfolg führt. Allerdings ist es für größere Probleme viel zu langsam. Heute sucht man nicht mehr irgendwelche Schnittebenen, sondern solche, die sich den Facetten des Polytops möglichst gut anschmiegen.

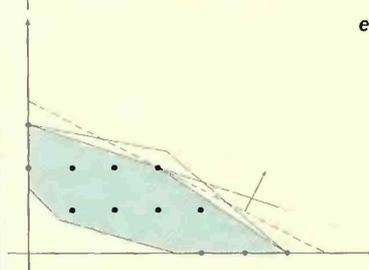
Durch Hinzunahme von Nebenbedingungen wird aus der unzulässigen Lösung des Odysseus-Rundreiseproblems im Kasten Seite 38 eine immer noch unzulässige, die jedoch längenmäßig der kürzesten zulässigen (der optimalen) Lösung bereits näher liegt (f), und im nächsten Schritt eine noch bessere, in der allerdings einige Kanten (gestrichelt) mit dem Wert 1/2 belegt sind (g). Der nächste Schritt liefert bereits die optimale Lösung (h).



c

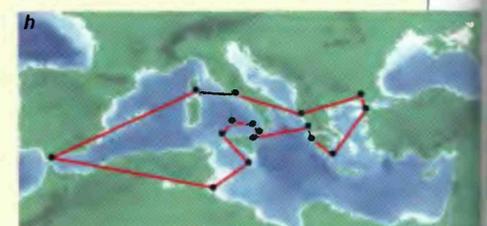
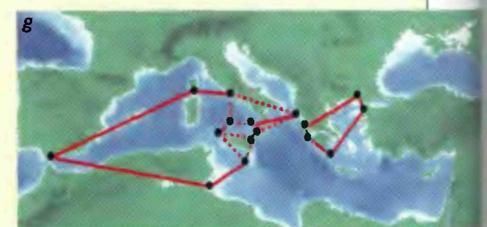
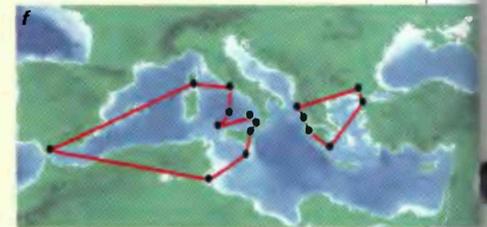


d



e

MARTIN GRÖTSCHHEL, MANFRED PADBERG



den Zielfunktionswert von 6859 Kilometer und den Lösungsvektor, welcher der optimalen Rundreise entspricht (*h*). Damit sind wir am Ziel.

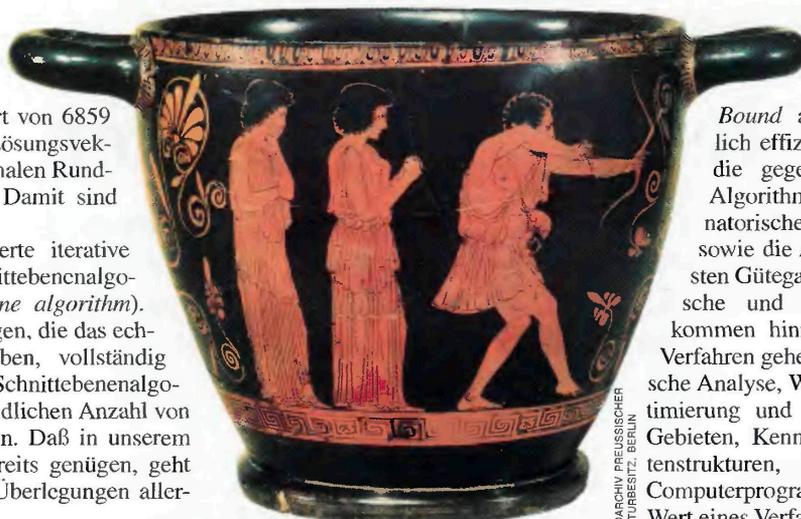
Das oben erläuterte iterative Verfahren heißt Schnittebenenalgorithmus (*cutting plane algorithm*). Sind die Ungleichungen, die das echte Polytop beschreiben, vollständig bekannt, muß der Schnittebenenalgorithmus nach einer endlichen Anzahl von Schritten beendet sein. Daß in unserem Fall vier Schritte bereits genügen, geht aus so allgemeinen Überlegungen allerdings nicht hervor.

Wir kennen gegenwärtig all diese Ungleichungen aber weder für das Travelling Salesman Problem noch für die meisten anderen kombinatorischen Optimierungsprobleme von praktischer Bedeutung. Ein Großteil der Forschung konzentriert sich darauf, dieses Wissen für immer mehr kombinatorische Probleme zu vergrößern. Dadurch sind die Grenzen mathematischer Berechenbarkeit weit über das hinaus getrieben worden, was sich Wissenschaftler noch vor wenigen Jahren vorstellen konnten.

Andererseits ist es möglich, daß die Schwierigkeit des TSP oder anderer kombinatorischer Optimierungsprobleme eine vollständige Kenntnis der erforderlichen Ungleichungen ausschließt. Wir wissen es einfach nicht; und gegenwärtig kennen wir mit Sicherheit nicht alle Ungleichungen. Anders als beim Problem des Odysseus kann also das iterative Schnittebenenverfahren beendet sein, bevor eine optimale Rundreise gefunden wurde.

Was sind unsere Wahlmöglichkeiten in einem solchen Fall? Wir könnten aufhören und uns mit der Kenntnis einer guten unteren Schranke begnügen. Wir könnten auch wie folgt ein Enumerationsverfahren anwenden: Wenn der Schnittebenenalgorithmus unschlüssig (also ohne eine zulässige Lösung für unser eigentliches Problem) endet, gibt es Kanten, die weder mit 0 noch mit 1 belegt sind. Wir greifen eine solche Kante heraus, setzen ihren Wert willkürlich auf 1 („die Tour muß diesen Weg verwenden“) und versuchen das derart eingeschränkte – und deshalb einfachere – Problem zu lösen. Dasselbe versuchen wir mit der Festlegung auf 0 statt 1 („die Tour darf diesen Weg nicht verwenden“). Von den beiden Lösungen, die sich dabei ergeben, nehmen wir die bessere. Eins von beiden – den Weg nehmen oder nicht – muß die Tour ja tun.

Natürlich kann der Lösungsversuch für diese Teilprobleme abermals unschlüssig enden. Wieder verzweigt der



BILDARCHIV FREIJSCHER  
KULTURBESTITZ, BERLIN

**Ganzzahlige Optimierung mit Pfeil und Bogen: Als Odysseus nach 20jähriger Irrfahrt wieder am heimischen Hof eintrifft, trennen ihn nur noch endlich viele unerfüllte Nebenbedingungen vom Ziel seiner Wünsche, nämlich die Freier Penelopes. Indem er gezielt eine Nebenbedingung nach der anderen wieder in Kraft setzt, eliminiert er sämtliche unzulässigen Punkte und erreicht dadurch endlich das Optimum (Darstellung auf einem attisch-rotfigurigen Skyphos).**

Lösungsprozeß in zwei Alternativen (*branching*), und es entsteht ein ganzer Baum von Lösungsmöglichkeiten. Den müssen wir auf möglichst geschickte Weise absuchen.

Das entsprechende Baumsuchverfahren ist unter dem Namen *Branch and Cut* bekannt. Es baut auf das in den sechziger Jahren entwickelte Verfahren *Branch and*



**Martin Grötschel (rechts) und Manfred Padberg** begannen ihre gemeinsame Arbeit an kombinatorischen Optimierungsproblemen 1974 an der Universität Bonn, wo Padberg Gastprofessor und Grötschel Doktorand war. Grötschel ist heute Professor für Mathematik an der Technischen Universität Berlin und Vizepräsident des Konrad-Zuse-Zentrums für Informationstechnik Berlin. Padberg war Gastprofessor an vielen bedeutenden Universitäten und ist heute Professor an der New York University. Für ihre Forschungsleistungen wurden die Autoren mit zahlreichen Preisen ausgezeichnet.

*Bound* auf, ist jedoch wesentlich effizienter. Auf ihm basieren die gegenwärtig erfolgreichsten Algorithmen für große kombinatorische Optimierungsprobleme sowie die Algorithmen mit den besten Gütegarantien. Diverse heuristische und enumerative Techniken kommen hinzu. In ein erfolgreiches Verfahren gehen gründliche mathematische Analyse, Wissen auf vielen der Optimierung und Informatik zugehörigen Gebieten, Kenntnis über effiziente Datenstrukturen, Datenverarbeitung und Computerprogrammierung ein. Denn der Wert eines Verfahrens erweist sich erst in der Durchführung auf realen Rechenanlagen.

Die hier erläuterten Beispiele für TSPs wurden mit einem Code berechnet, den Manfred Padberg und Giovanni Rinaldi vom italienischen Forschungszentrum IASI (Istituto di Analisi dei Sistemi ed Informatica) in Rom entwickelt haben. Es gibt weit mehr erfolgreiche Anwendungen von *Branch and Cut* durch uns und andere, als in diesem Artikel beschrieben werden können. Die Lösungszeiten für die verschiedenen Fragestellungen eines 16-Städte-Problems betragen nur wenige Millisekunden auf einer modernen Workstation. Man vergleiche diese Zeit mit den 92 Stunden für die stumpfsinnige Enumeration! Dazu der rasante Leistungszuwachs der Rechner-Hardware, solide angewandte Mathematik und intelligente Software-Entwicklung: Damit kann eine exakte Lösung von weitaus größeren Problemen, als wir uns jetzt vorstellen können, leicht zur Realität werden.

Zum Schluß noch einmal die Frage: Warum ist das Travelling Salesman Problem so schwer? Beim beschriebenen Lösungsverfahren ist die astronomisch hohe Zahl von Ungleichungen, die zu berücksichtigen sind, zweifellos die größte Hürde. Liegt darin die prinzipielle Schwierigkeit des Problems begründet? Dies ist sicherlich teilweise der Fall. Erstaunlicherweise ist dies aber keine vollständige Erklärung. Es gibt Probleme in der kombinatorischen Optimierung, die sowohl in der Theorie als auch in der Praxis leicht zu lösen sind, deren Polytope aber bei gleicher Dimension weit mehr Ungleichungen als das zum TSP gehörige Polytop benötigen.

Wenn es also weder die Anzahl möglicher Lösungen noch die Anzahl der erforderlichen Ungleichungen ist, was ist es dann, was ein Problem „schwierig“ macht? Gegenwärtig kennt niemand eine Antwort auf diese Fragen; gleichwohl liefern die hier vorgestellten Algorithmen gute Ergebnisse. ■