Das Knotenüberdeckungsproblem

Eine Fallstudie zur Didaktik NP-schwerer Probleme

Teil 1

von Rolf Niedermeier, Jörg Vogel, Michael Fothe und Mirko König

In diesem Beitrag wird das Thema „NP-schwere Probleme“ für die Schule aus didaktisch-methodischer Sicht aufbereitet. Stellvertretend geschieht dies am Knotenüberdeckungsproblem, das erhebliche praktische Relevanz besitzt. Ausgehend von unterschiedlichen Modellierungen werden verschiedene Lösungsverfahren erläutert und bewertet. Die mit NP-schweren Problemen verbundene, anscheinend unvermeidliche kombinatorische Explosion wird thematisiert. Ausgehend vom konkreten Problem werden Problemverwandtschaft, Determinismus und Nichtdeterminismus intuitiv erklärt. Dies mündet in Erläuterungen zur P=NP-Frage, einem der interessantesten Probleme der aktuellen Forschung in der Informatik.

P = NP?

Uns ist kein wissenschaftliches Problem bekannt, das „allumfassender“ und dennoch einfacher zu formulieren wäre als die berühmte P=NP-Frage: Grob gesagt geht es hierbei darum zu ergründen, ob eine Vielzahl natürlicher und anwendungsbezogener Probleme algorithmisch effizient gelöst werden kann, d. h. ob die Zeitkomplexität dieser Probleme sich im Verhältnis zur Länge der Eingabe polynomial verhält (siehe Kasten „P und NP“, nächste Seite).

Erstmals wurde die P=NP-Frage 1971 von Stephen Cook und Leonid Levin gestellt – und seitdem sind außergewöhnliche Forschungsanstrengungen unternommen worden, um dieses Problem zu lösen. Das Besondere hierbei ist, dass zum einen schon Tausende von Forschern aus völlig verschiedenen Gebieten von der Soziologie über die Ingenieurwissenschaften und die Biologie bis zur Energieforschung sich mit dieser Fragestellung beschäftigt haben, andererseits die zugrunde liegende Thematik sehr einfach und intuitiv vermittelbar ist. Wir sind der Auffassung, dass die P=NP-Frage mit all ihren Facetten so wichtig ist, dass sie schulisches Allgemeingut sein sollte. Mit diesem Beitrag wollen wir versuchen, die elementare Faszination, die von sicherlich einer der bedeutendsten wissenschaftlichen Fragestellungen der Informatik ausgehen kann, an einem einfachen und wichtigen Problem herauszuarbeiten.

Das in diesem Beitrag vorgestellte Problem befasst sich mit der Überdeckung der Kanten eines Graphen mit möglichst wenigen Knoten und wird als *Knotenüberdeckungsproblem* bezeichnet. Anders als frühere Artikel zur P=NP-Frage wollen wir uns hiermit also ausschließlich auf ein konkretes und einfach zu formulierendes Problem konzentrieren und mit seiner Hilfe eine möglichst ursprüngliche Begeisterung wecken, ohne allzu formalistisch werden zu müssen.

Beispiele zur Knotenüberdeckung in Graphen

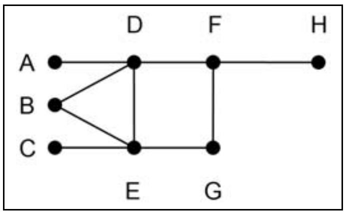
Viele Probleme aus Alltag und Wissenschaft lassen sich als Knotenüberdeckungsprobleme formulieren. Die nachfolgend genannten Fallbeispiele verdeutlichen am konkreten Problem *Knotenüberdeckung*, in welch verschiedenen Bereichen das mit der P=NP-Frage inhärent verknüpfte Phänomen der kombinatorischen Explosion auftreten kann.

Ein Partyproblem

Nehmen wir an, Person X will eine harmonische Geburtstagsfeier gestalten. Einzuladende Kandidaten wären am liebsten acht Personen, nämlich die Personen A bis H. Nun ist es jedoch so, dass nicht alle potenziellen Gäste miteinander harmonieren; Ziel von X ist es, möglichst viele der acht Personen so einzuladen, dass nur miteinander harmonierende Personen an der Feier teilnehmen.

Man versuche, eine möglichst gute Lösung zu finden, wenn Nicht-Harmonie genau zwischen folgenden Personenpaaren besteht: (A,D), (B,D), (B,E), (C,E), (D,F), (D,E), (E,G), (F,G), (F,H).

Eine grafische Darstellung dieses Beziehungsgeflechts sieht wie nebenstehend aus:



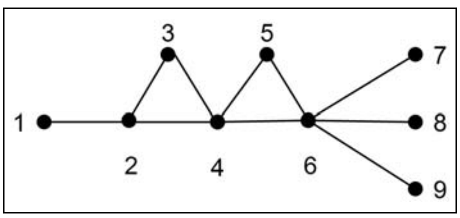
X kann bis zu fünf Personen für eine harmonische Party einladen.

Ein Experimentproblem

Forscherin A arbeitet in einer experimentellen Wissenschaft auf Basis teurer, nicht immer fehlerfreier Experimentdurchführungen an einer neuen wissenschaftlichen Hypothese. Leider widersprechen sich einige der Experimentergebnisse, was aber allein an der Fehlerhaftigkeit der Ergebnisdaten liegen kann. In den empirischen Wissenschaften ist es daher nicht unüblich, folgende Plausibilitätsbetrachtung anzustellen: Die aus den Experimentdaten abzuleitende Hypothese ergibt sich am besten bzw. am wahrscheinlichsten so, indem man eine möglichst kleine Zahl von (vermutlich) fehlerhaften Experimentergebnissen eliminiert, sodass der verbleibende Rest in sich widerspruchsfrei ist und somit zur Hypothesenbildung taugt.

Man versuche eine möglichst breite Hypothesenbasis zu finden, wenn folgende Experimentpaare jeweils einander widersprechen: (1,2), (2,3), (2,4), (3,4), (4,5), (4,6), (5,6), (6,7), (6,8), (6,9).

Nebenstehend wieder eine grafische Darstellung:



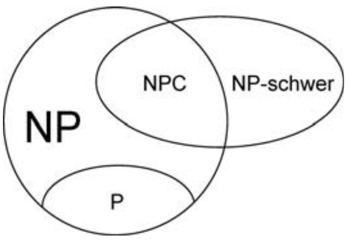
Die Elimination von drei der neun Experimente würde zu einer schlüssigen Hypothese führen.

Ein Verkehrsüberwachungsproblem

Im Zuge eines Großereignisses in einer Stadt soll der Verkehrsfluss effizient gesteuert werden. Hierzu ist es wichtig, dass jeder Straßenzug zwischen zwei Straßenkreuzungen wenigstens von einer „Kreuzung“ aus eingesehen werden kann. Dies sei genau dann der Fall, wenn an mindestens einem der beiden Enden des Straßenzugs ein Verkehrspolizist stationiert ist.

P und NP

In der Komplexitätstheorie werden Probleme klassifiziert, deren Lösungen von Informatiksystemen bzw. Computern berechnet werden können, und zwar gemäß dem für die Berechnung erforderlichen Aufwand. Programme und Algorithmen, die in der Praxis verwendet werden, sollten möglichst effizient arbeiten, d. h. sie sollten ein vorgegebenes Problem in möglichst kurzer Zeit oder mit möglichst geringem Einsatz von Betriebsmitteln (vor allem Speicherplatz) lösen.
Ein mögliches quantitatives Maß des Berechnungsaufwands ist die Anzahl der maximal notwendigen Rechenschritte in Bezug auf die Größe der Eingabe (Zeitkomplexität). Lässt sich die Zeitkomplexität als Polynom der Form n^k angeben, wobei n die Größe der Eingabe ist, so gehört das Problem zur Klasse P (von ***p**olynomial*) und gilt somit als „praktisch lösbar“. „Nicht praktisch lösbar“ sind die Probleme mit exponentieller Zeitkomplexität.
Die Komplexitätsklasse NP (von ***n**ichtdeterministisch **p**olynomial*) ist die Menge aller nichtdeterministisch in Polynomialzeit lösbaren Probleme. Bei einem deterministischen Algorithmus ist der Programmablauf eindeutig bestimmt. Bei einem nichtdeterministischen Algorithmus dagegen besteht bezüglich des auf einen Rechenschritt folgenden Schritts eine Wahlmöglichkeit.
Die Klasse P ist eine Teilmenge der Klasse NP, da jedes deterministisch in Polynomialzeit lösbare Problem auch nichtdeterministisch in Polynomialzeit lösbar ist.
Ein Problem heißt „NP-vollständig“, wenn es in der Klasse NP liegt und wenn sich seine Komplexität um nicht mehr als ein Polynom von der Komplexität jedes anderen Problems in NP unterscheidet. Die NP-vollständigen Probleme (Klasse NPC; von ***c**omplete*) sind also die schwierigsten Probleme in NP.



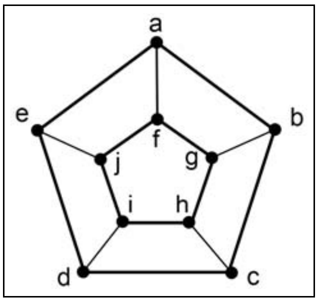
Zusammenhang zwischen den Komplexitätsklassen, falls P ≠ NP.

Es ist ein zentrales offenes Problem der Informatik, ob P=NP oder P≠NP ist, d. h. ob mit deterministischen Algorithmen in Polynomialzeit genau so viele oder weniger Probleme gelöst werden können wie mit nichtdeterministischen Algorithmen.
Der Beweis von P=NP würde bedeuten, dass für Probleme der Klasse NP Algorithmen existieren, die eine Lösung in – wesentlich geringerer – Polynomialzeit erzeugen können.

Ziel ist es, Einsicht in das gesamte Verkehrsnetz mit all seinen Straßenzügen zu haben und dies mit möglichst wenigen Verkehrspolizisten zu erreichen. Betrachten Sie folgendes System von fünfzehn Straßen, jeweils repräsentiert durch ihre beiden „Kreuzungsendpunkte“:

(a,b), (a,e), (a,f), (b,c), (b,g), (c,d), (c,h), (d,e), (d,i), (e,j), (f,g), (f,j), (g,h), (h,i), (i,j).

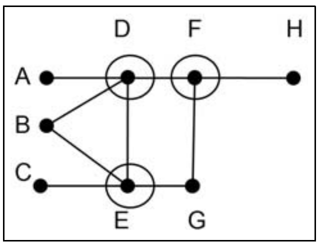
Auch dies lässt sich grafisch repräsentieren:

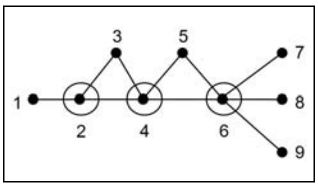


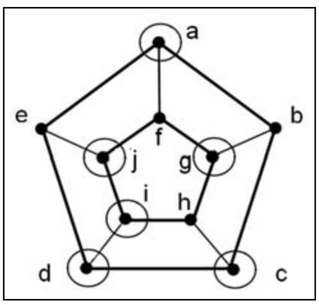
Es gibt eine Lösung mit sechs Verkehrspolizisten.

Zu Knotenüberdeckungen

Alle dargestellten Beispiele fallen unter das Paradigma der Konfliktauflösung. Sei es die Entscheidung für bzw. gegen einen Partyteilnehmer, sei es die Entscheidung für bzw. gegen die Miteinbeziehung von Experimentergebnissen zur Hypothesenbildung oder die Entscheidung zwischen verschiedenen Kreuzungspositionen zur Stationierung von Verkehrspolizisten. Diese verschiedenen Szenarien führen letztlich stets auf dieselbe Problemstellung: *Knotenüberdeckung in ungerichteten Graphen*. Die Probleme lassen sich mit folgenden Graphen modellieren, anhand derer die Lösungssuche für einen Menschen stark vereinfacht wird:







Die zu entfernenden Knoten sind leicht zu finden: Lösungsmengen sind z. B. {D, E, F} bzw. {2, 4, 6} bzw. {a, c, d, g, i, j}. Es gibt jeweils weitere Lösungen.

Das zugrunde liegende Problem der Knotenüberdeckung (engl.: *vertex cover* oder *node cover* – im Übrigen hätte man das Problem genauso gut Kantenüberdeckung nennen können) lässt sich also wie folgt formalisieren:

▷ *Eingabe:*
Ein ungerichteter Graph G = (V, E) und eine positive ganze Zahl k.
V – set of vertices = Knotenmenge,
E – set of edges = Kantenmenge.

▷ *Frage:*
Gibt es eine Teilmenge von Knoten $U \subseteq V$, sodass jede Kante aus E mindestens einen ihrer beiden Endknoten in U hat, wobei U nicht mehr als k Elemente hat?

Man spricht also davon, dass die Kanten von Knoten überdeckt (engl.: *to cover*) werden. Der Graph besitzt dann eine Knotenüberdeckung.

Das Knotenüberdeckungsproblem ist eines der am längsten bekannten und elementarsten Probleme aus der Menge der sogenannten *NP-vollständigen Probleme*. In der wissenschaftlichen Fachliteratur ist dieses Problem zusammen mit diversen Varianten vielfach untersucht worden. So findet sich auch eine Vielzahl von Anwendungsszenarien für Knotenüberdeckungen, die neben den oben genannten von der Bioinformatik bis hin zu Netzwerküberwachung reichen.

Insbesondere lassen sich an diesem einfach zu verstehenden Problem zentrale Ideen, Konzepte und Zielsetzungen rund um die P=NP-Frage illustrieren. Dies ist Gegenstand der nachfolgenden Betrachtungen.

Wir weisen darauf hin, dass das Knotenüberdeckungsproblem, das oben als Entscheidungsproblem formalisiert wurde, auch als Optimierungsproblem formuliert werden kann:

Gegeben ist ein Graph G = (V, E). Finden Sie eine kleinste Teilmenge $U \subseteq V$ von Knoten, die eine Überdeckungsmenge ist.

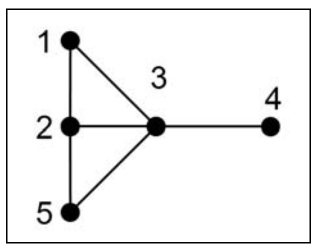
Ein solches Optimierungsproblem ist im Gegensatz zum Entscheidungsproblem nicht NP-vollständig, wohl aber NP-schwer, d. h. mindestens so schwer wie die NP-Probleme.

Modellierungsaspekte und Problemverwandtschaft

Graphproblem

Das Knotenüberdeckungsproblem wurde als *Graphproblem* eingeführt. Es ist jedoch für das Problemverständnis wichtig und eventuell für Lösungsansätze hilfreich zu sehen, wie ein und dasselbe konkrete Graphproblem auf weitere Weisen modelliert werden kann. Dies wollen wir hier zunächst näher betrachten.

Von Graphen kommt man vermöge des Adjazenzmatrixbegriffs unmittelbar zu einem *0-1-Matrixproblem*. Betrachten wir dazu nebenstehenden Graphen mit seiner zugehörigen Adjazenzmatrix, die die Nachbarschaftsverhältnisse ausdrückt und die aus diesem Grund symmetrisch ist, mit der Hauptdiagonalen als Symmetrieachse. Eine solche Matrix lässt sich als Dreiecksmatrix darstellen. (Eine 1 steht für: es gibt die Kante zwischen den Knoten x und y; eine 0 steht für: es gibt keine Kante.)



Betrachtet man das Knotenüberdeckungsproblem in seiner „Matrixform“, so geht es darum, möglichst wenige Indizes so zu markieren, dass alle Einsen in der Matrix von den markierten Zeilen und Spalten überdeckt werden. Im Beispiel überdecken die Zeilen und Spalten mit den Indizes 2 und 3 alle Einsen. Das heißt, die Kanten von den Knoten 2 und 3 weisen zu allen anderen Knoten des Graphen.

	1	2	3	4	5
1					
2	1				
3	1	1			
4	0	0	1		
5	0	1	1	0	

Eine weitere äquivalente Formulierung des Knotenüberdeckungsproblems ist durch die Darstellung als *Mengenproblem* gegeben, wobei die Kanten als zwei-elementige Mengen interpretiert werden. Obiger Graph führt zu folgender Menge von Mengen über der Grundmenge S = {1,2,3,4,5} : {{1,2}, {1,3}, {2,3}, {2,5}, {3,4}, {3,5}}.

Ziel ist es nun, möglichst wenige Elemente aus S so auszuwählen, dass jede zwei-elementige Menge mindestens eines dieser Elemente enthält. Dieses Problem ist in der Fachliteratur als *Hitting-Set-Problem* bekannt.

Diese Hitting-Set-Sichtweise auf Knotenüberdeckung führt sogleich zu einer weiteren Interpretation als Matrixproblem bzw. auch als *geometrisches Problem*. Man betrachte eine Matrix, in der die Spalten die Menge S und die Zeilen die zwei-elementigen Mengen repräsentieren. In unserem Beispiel sieht das so aus:

	1	2	3	4	5
{1,2}	X	X			
{1,3}	X		X		
{2,3}		X	X		
{2,5}		X			X
{3,4}			X	X	
{3,5}			X		X

Ein Kreuz stellt also das Vorkommen eines Elements in einer Menge dar. Eine gesuchte Lösung besteht nun aus einer kleinstmöglichen Menge von Spalten (geometrisch gesprochen: senkrechten Strichen), sodass jede Zeile in wenigstens einem Kreuz getroffen wird.

En passant weisen wir auf einen wichtigen, effizient lösbaren Spezialfall hin, der in dieser Modellierung deutlich zutage tritt: Falls es möglich ist, die Spalten so zu vertauschen, dass in jeder Zeile alle Kreuze lückenlos nebeneinander stehen, so ist das Problem im Gegensatz zum allgemeinen Fall effizient lösbar. In der Literatur spricht man in diesem Spezialfall von der *Consecutive Ones Property*.

Eine weitere Modellierungsmöglichkeit für das Knotenüberdeckungsproblem bietet die mathematische Logik: Man betrachte folgende aussagenlogische Formel in „2-konjunktiver Normalform“:

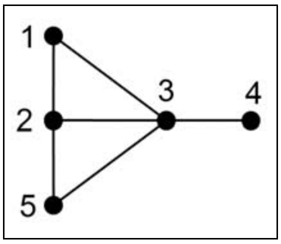
$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_2 \vee x_5) \wedge (x_3 \vee x_4) \wedge (x_3 \vee x_5)$.

Hierbei entspricht jede aussagenlogische Variable x_i genau dem Knoten i und jede Klausel $(x_i \vee x_k)$ genau der Kante {i, k}. Die Formel ist offensichtlich erfüllbar. Ziel ist es nun jedoch, eine die Formel erfüllende Wahrheitswertbelegung der Variablen zu finden, sodass möglichst wenige Variablen auf WAHR gesetzt werden müssen. Eine solche Belegung wäre z. B. $x_2 = x_3$ = WAHR; die übrigen Variablen können dann den Wert FALSCH annehmen.

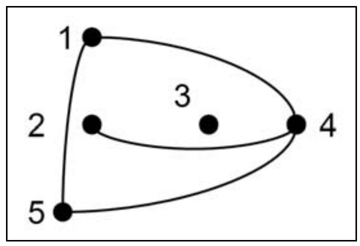
Zusammenfassend ist festzuhalten, dass es verschiedene äquivalente Formulierungen des Knotenüberdeckungsproblems gibt. Insbesondere heißt das, dass Übersetzungen der Problemformulierungen ineinander (sogenannte Reduktionen) existieren, die im Wesentlichen immer das gleiche Problem darstellen.

Verwandte Probleme

Nun wollen wir einige mit dem Knotenüberdeckungsproblem verwandte Probleme betrachten. Wir beginnen mit dem *Problem der unabhängigen Mengen*: Eine unabhängige Menge in einem Graph G = (V, E) ist eine Knotenmenge derart, dass für jedes Paar von Knoten aus dieser Knotenmenge gilt, dass keine Kante zwischen ihnen existiert. Das Gegenteil einer unabhängigen Menge ist eine sogenannte Clique, welche einen vollständigen (Teil-)Graphen bildet: jeder Knoten ist mit jedem anderen Knoten der Clique durch eine Kante verbunden. Betrachten wir folgendes Beispiel:



und der „Komplementgraph“ dazu:



Der Komplementgraph entsteht aus einem gegebenen Graphen, in dem man alle Kanten löscht und eine neue Kante genau zwischen solchen Knotenpaaren einfügt, wo bisher keine Kante bestand.

Es gilt: Ein Graph G = (V, E) mit n Knoten besitzt eine Knotenüberdeckung mit k Knoten genau dann, wenn er eine unabhängige Menge mit n – k Knoten besitzt genau dann, wenn sein Komplementgraph einen vollständigen Teilgraphen mit n – k Knoten besitzt. In obigem Beispiel besitzt der linke Graph die Knotenüberdeckungsmenge {2, 3} und die unabhängige Menge {1, 4, 5} und sein rechts gezeichneter Komplementgraph die Clique {1, 4, 5}.

Somit folgt aus der NP-Vollständigkeit des Knotenüberdeckungsproblems unmittelbar die der anderen beiden Probleme, wenn alle drei Probleme entsprechend als Entscheidungsprobleme formuliert sind. Löst man nämlich eines der Probleme durch einen effizienten Algorithmus, so kann dieser auch mit leichter Modifikation zur effizienten Lösung der anderen benutzt werden.

Nicht immer ist die Übersetzung (Reduktion) zwischen NP-vollständigen Problemen so leicht zu sehen. Betrachten wir z. B. das *Problem der dominierenden Mengen* in Graphen: Gegeben ist ein ungerichteter Graph G = (V, E) und eine positive natürliche Zahl k.

NP-schwere Probleme in den Lehrplänen

Nach den Einheitlichen Prüfungsanforderungen in der Abiturprüfung (EPA) ist die Behandlung des Themas „NP-schwere Probleme“ in allen Bundesländern Deutschlands möglich. In den einzelnen Lehrplänen ist der direkte Bezug zu diesem Gegenstand jedoch unterschiedlich ausgeprägt. Anhand von Lehrplanschwerpunkten und Zitaten soll in der folgenden Tabelle die Suche nach der richtigen Stelle im Lehrplan erleichtert werden.

Tabelle der Bundesländer

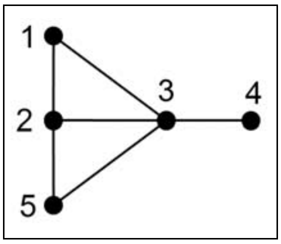
	Landeslehrplan, Jahr	**Lehrplaneinheit oder Zitat aus dem Lehrplan**
1	Baden-Württemberg Lehrplan für das Fach Informatik in der Kursstufe des Gymnasiums 2001	Lehrplaneinheit 7: Praktische und theoretische Grenzen des Rechnereinsatzes
2	Bayern EGy III (G8 liegt noch nicht vor)	Die Schüler lernen die prinzipiellen Grenzen informatischer Systeme kennen und erhalten Einblicke in die Problemkreise der Korrektheit und Komplexitätsabschätzung von Informatiksystemen
3	Berlin Rahmenlehrplan für die gymnasiale Oberstufe 2006	4.5 Informatik, Mensch und Gesellschaft Eine direkte Einordnung NP-schwerer Probleme ist nicht vorgegeben, aber grundsätzlich möglich
4	Brandenburg Rahmenlehrplan für den Unterricht in der gymnasialen Oberstufe 2006	Die Schülerinnen und Schüler … beurteilen die Grenzen des Einsatzes von Informatiksystemen … Eine direkte Einordnung NP-schwerer Probleme ist nicht vorgegeben, aber grundsätzlich möglich.
5	Bremen Fachrahmenplan Informatik Gymnasiale Oberstufe 2001	Grundlagen der Theoretischen Informatik
6	Hamburg 2004	Die Schülerinnen und Schüler … • schätzen die Zeitkomplexität von Algorithmen ab • ordnen Probleme Komplexitätsklassen zu
7	Hessen 2006	13.1 Konzepte und Anwendungen der Theoretischen Informatik
8	Mecklenburg-Vorpommern 2001	Algorithmisch lösbare und unlösbare Probleme
9	Niedersachsen 2007	Das Kultusministerium nimmt lediglich Bezug auf die EPA
10	Nordrhein-Westfalen 1998	Ausgewählte Gebiete der theoretischen Informatik
11	Rheinland-Pfalz 2004	Grenzen algorithmisch arbeitender Systeme Die Existenz praktisch nicht durchführbarer algorithmischer Problemlösungen aufzeigen
12	Saarland 1997	Unterrichtsthema 1.2: Praktische Grenzen der Berechenbarkeit
13	Sachsen 1992	Wahlgrundkurs 12/II: Gesellschaftliche und theoretische Probleme der Informatik – Lernbereich 3: Algorithmentheorie (14 Std.)
14	Sachsen-Anhalt 2003	Thema: Informatik und Gesellschaft Inhalte: Grenzen von Informatiksystemen Eine direkte Einordnung NP-schwerer Probleme ist nicht vorgegeben, aber grundsätzlich möglich
15	Schleswig-Holstein 2002	4.4.2 Jahrgangsstufe 12.1: Algorithmen und Datenstrukturen Algorithmen und Datenstrukturen werden zu ausgewählten Problemen entwickelt und durch Effizienzbetrachtungen hinsichtlich ihrer Brauchbarkeit untersucht
16	Thüringen 1999	Themenbereich 8: Möglichkeiten und Grenzen des Einsatzes von Informatiksystemen

Auszug aus den EPA Informatik

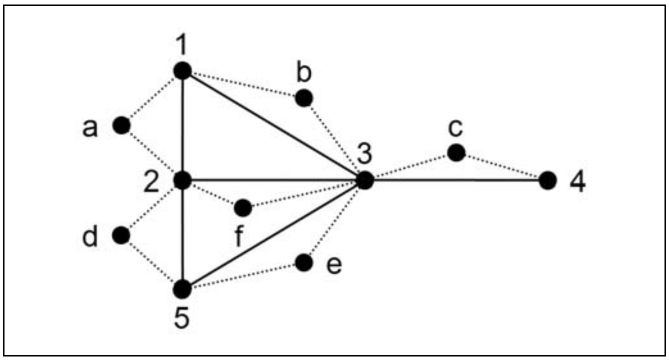
„Aus dem Unterricht ist bekannt, dass es Probleme gibt, die prinzipiell mit einem Computer unlösbar sind oder bei denen bei bekanntem Lösungsalgorithmus die verfügbaren Ressourcen zur Problemlösung nicht ausreichen und dass zahlreiche dieser Probleme hohe praktische Relevanz besitzen. Im Unterricht wurden Algorithmen entworfen und die Zeitkomplexität von Algorithmen abgeschätzt. Der Entwurf wurde in Beschreibungs-, Strukturierungs- und algorithmische Phase gegliedert. Die Auswirkungen des Computereinsatzes in mehreren gesellschaftlichen Bereichen wurde diskutiert.“

Quelle: KMK – Ständige Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland: Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik. Beschluss der Kultusministerkonferenz vom 01.12.1989 i. d. F. vom 05.02.2004. S. 46 (http://www.kmk.org/doc/beschl/EPA-Informatik.pdf).

Frage: Gibt es eine Teilmenge von Knoten $D \subseteq V$, sodass jeder andere Knoten wenigstens einen Nachbarn in dieser dominierenden Menge und D nicht mehr als k Elemente hat? Hier ist wieder ein Beispiel:



und



Der obere Graph besitzt die Knotenüberdeckung {2, 3} und die dominierende Menge {3}. Ersetzt man nun im oberen Graphen jede Kante durch einen Dreiecksgraphen, was zum unteren Graphen mit den zusätzlichen Knoten a, b, c, d, e, f führt, so gilt folgende Aussage: Der obere Graph hat eine Knotenüberdeckung der Größe k genau dann, wenn der untere Graph eine dominierende Menge der Größe k hat. Diese Aussage gilt für beliebige Graphen bei Anwendung der beschriebenen „Dreieckstransformation“. Bemerkenswert ist nun, dass es ungleich schwerer ist, die umgekehrte Richtung – d.h. eine Reduktion des Dominierungsproblems auf das Knotenüberdeckungsproblem – zu finden.

Ein Verfahren zur Lösung des Knotenüberdeckungsproblems

Für die bisherigen einführenden Beispiele war die Beantwortung der Frage nach der Existenz einer Knotenüberdeckungsmenge der Größe k durch einen geschulten Blick oder durch eine vollständige Fallunterscheidung möglich. Warum erweist sich das Knotenüberdeckungsproblem dennoch als ein zwar algorithmisch grundsätzlich beherrschbares, aber in der Praxis schwierig zu lösendes Problem? Allen bisherigen Beispielen ist gemeinsam, dass die zugehörigen Graphen nur wenige Knoten und damit wenige Kanten besitzen. Je mehr Knoten und Kanten ein gegebener Graph hat, desto schwieriger wird es im Allgemeinen, die Frage nach der Existenz einer Knotenüberdeckungsmenge der Größe k zu beantworten.

Dabei ist es eine einfache Aufgabe zu entscheiden, ob eine gegebene Teilmenge U der Knotenmenge V eine Knotenüberdeckungsmenge des Graphen ist:

> Es wird eine Liste aus der Kantenmenge E des Graphen benötigt. Eine solche Liste ist entweder direkt gegeben oder sie lässt sich einfach aus der Adjazenzmatrix des Graphen erstellen. Für jede Kante e = {u, v} wird getestet, ob (mindestens) einer ihrer Endknoten u oder v zur Menge U gehört. Wenn der gegebene Graph G = (V, E) n Knoten und m Kanten besitzt, dann ist für jedes vorgegebene U der Test garantiert mit maximal $n \cdot m$ Prüfschritten möglich: Denn für jede der m Kanten wird maximal die Menge der n Knoten geprüft.

Ein allgemeiner Ansatz zur Lösung des Knotenüberdeckungsproblems besteht darin, *systematisch* eine Liste aller Teilmengen der Knotenmenge V zu erzeugen und dann für jede erzeugte Teilmenge zu testen, ob sie eine Knotenüberdeckungsmenge ist. Ein solches Verfahren der *vollständigen Durchmusterung* aller Teilmengen setzt eine Nummerierung der Knotenmenge V voraus und liefert mit Sicherheit eine kleinstmögliche Knotenüberdeckungsmenge, aber es hat einen großen Nachteil: Es erfordert einen riesigen Zeitaufwand im schlechtesten Fall (engl.: *worst case*).

Die Knotenmenge V mit n Knoten besitzt 2^n verschiedene Teilmengen, d.h. in Abhängigkeit von der Knotenzahl n des Graphen G sind exponentiell viele Teilmengen (d.h. exponentiell viele Fälle) zu testen. Bei jeder beliebigen Reihenfolge ist es möglich, dass die gesuchte Menge gerade die letzte in dieser Reihenfolge ist.

Auch das Testen von „Stichproben“, d.h. willkürlich oder zufällig herausgegriffenen Mengen nutzt nichts: Die Antwort „nein – keine Knotenüberdeckungsmenge“ sagt nichts über die korrekte Antwort der Nachbarmengen in der gegebenen Reihenfolge!

So sehr man sich auch bemühte: Es wurde kein Verfahren zur präzisen Lösung des Problems gefunden, das letztlich die Durchmusterung sämtlicher Teilmengen grundsätzlich vermeidet.

Für einen Graphen G mit n Knoten sind also im schlechtesten Fall 2^n Teilmengen zu testen, wobei für jede einzelne Teilmenge der Aufwand durch $n \cdot m$ Rechenschritte beschränkt und damit durchaus vertretbar ist (m bezeichnet hierbei die Anzahl der Kanten des Graphen G).

Unsere Beispiele sind Graphen mit etwa 10 Knoten. Also gibt es $2^{10} \approx 1000$ zu testende Teilmengen. Ein Mensch wird diese 1000 Fälle nicht wirklich untersuchen müssen. Ihm helfen Anschauung und Intuition.

Genau dies hat ein Computer nicht, und er wird in systematischer Weise alle Fälle durchmustern. Dabei sind 1000 Fälle kein Problem für einen heutigen Computer.

Anders bei praktisch relevanten Aufgabenstellungen mit etwa 100 Knoten: Hier wären in systematischer

Weise 2^{100} Fälle zu untersuchen. Nehmen wir an, dass ein Rechner zur Verfügung steht, der pro Sekunde eine Million Fälle testen kann. Dann würde ein solcher Computer dennoch etwa $3 \cdot 10^{16}$ Jahre rechnen müssen, was ein Vielfaches des Lebensalters des Universums ist.

Genau dieser überraschende Effekt wird als *kombinatorische Explosion* bezeichnet. Er besteht darin, dass der Aufwand zur Lösung eines gegebenen Problems der Größe n exponentiell in n wächst. Das Rechenbeispiel zeigt: Probleme, die einen derartigen Aufwand erfordern, erweisen sich als praktisch unlösbar. Da hilft auch nicht das Warten auf leistungsfähigere Computertechnik: Ein 1000-mal schnellerer Computer gestattet es in diesem Beispiel lediglich, eine um 10 Knoten vergrößerte Eingabe in der gleichen Zeit zu bearbeiten, da $2^{10} \approx 1000$ gilt.

Wir fassen die Beobachtungen für das Knotenüberdeckungsproblem zusammen:

▷ Für eine einzelne Teilmenge U der Knotenmenge V ist der Aufwand für den Test, ob U eine Knotenüberdeckungsmenge ist, in akzeptabler Zeit möglich.
▷ Die kombinatorische Explosion ergibt sich aus der exponentiellen Anzahl der Fälle, die zu untersuchen sind, und macht eine präzise Lösung des Problems praktisch für nicht ganz kleine Knotenmengen unrealisierbar.

Trotz vielfältiger Bemühungen wurden keine Lösungsverfahren entdeckt, die den exponentiellen Aufwand vermeiden.

(Fortsetzung im nächsten Heft)

Rolf Niedermeier
Jörg Vogel
Michael Fothe
Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
07740 Jena

Mirko König
Carl-Zeiss-Gymnasium Jena
Erich-Kuithan-Straße 7
07743 Jena

E-Mail:
{niedermr/vogel/fothe/mkoenig}@minet.uni-jena.de

Eine ausführliche Literaturliste wird im folgenden zweiten Teil dieses Beitrags vorgestellt.