



**JENAER SCHRIFTEN ZUR
MATHEMATIK UND INFORMATIK**

Eingang: 19. Juli 2006 Math / Inf / 09 / 06 Als Manuskript gedruckt

**Das Knotenüberdeckungsproblem -
eine Fallstudie zur Didaktik NP-schwerer Probleme**

Rolf Niedermeier¹, Jörg Vogel¹, Michael Fothe¹, Mirko König²

¹Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
07740 Jena

²Carl-Zeiss-Gymnasium Jena
Erich-Kuithan-Str.7
07743 Jena

{niedermr,vogel,fothe,mkoenig}@minet.uni-jena.de

Zusammenfassung:

Mit dem Artikel 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 anschaulich thematisiert. Ausgehend vom konkreten Problem werden Problemverwandtschaft, Determinismus und Nichtdeterminismus intuitiv erklärt. Dies mündet in Erläuterungen zum $P =? NP$ -Problem, einem der interessantesten Probleme der aktuellen Forschung. Das Thema wird anhand des Thüringer Lehrplans in den Informatikunterricht eingeordnet.

1. Einleitung

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, d. h. in Laufzeit polynomial wachsend in der Eingabegröße gelöst werden kann. Das Besondere hierbei ist, dass zum einen schon Tausende von Forschern aus verschiedensten Gebieten von der Soziologie über die Ingenieurwissenschaften und die Biologie bis zur Energieforschung auf die Fragestellung stießen, andererseits die zugrundeliegende Thematik sehr einfach und intuitiv vermittelbar ist. Wir sind der Auffassung, dass das P - NP -Thema mit all seinen Facetten so wichtig ist, dass es schulisches Allgemeingut sein sollte. Mit diesem Artikel wollen wir versuchen, die elementare Faszination, die von sicherlich einer der zentralsten wissenschaftlichen Fragestellungen überhaupt ausgehen kann, an einem einfachen und wichtigen Problem herauszuarbeiten.

Das in dieser Arbeit studierte 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 für diese faszinierende Welt der Wissenschaft wecken, ohne sehr formalistisch werden zu müssen.

Dieser Artikel ist wie folgt gegliedert: Zunächst werden wir das besagte Knotenüberdeckungsproblem anhand einiger konkreter Anwendungen motivieren. Dahinter steckt die auch allgemein sehr wichtige Zielsetzung, sich in der mathematisch-informatischen Modellierung von Problemen zu üben. Das „Gleiche im vermeintlich Verschiedenen“ zu entdecken ist hier von zentraler Bedeutung. Nachdem das Problem motiviert, definiert und modelliert wurde, setzen wir an, die algorithmische Schwierigkeit des Problems zu erfassen. Kernthema hierbei ist ein Grundverständnis für anscheinend unvermeidbare kombinatorische Explosionen zu entwickeln. Der Schatz an Fragestellungen im Umkreis von P versus NP ist ungeheuer reichhaltig und nur selten gibt es hier erschöpfende Antworten und insbesondere bedarf im Allgemeinen jedes kombinatorisch explosive Problem eigener Lösungsideen. Exemplarisch verdeutlichen wir das hier anhand des Knotenüberdeckungsproblems. Dies leitet dann über zu einer etwas präziseren, aber doch informellen Fassung der $P = ? NP$ -Frage. Wir werden hier einen knappen Überblick über den Stand der Diskussion in der Wissenschaft geben. Abschließend kehren wir zu den bestehenden Lösungsmöglichkeiten für das Knotenüberdeckungsproblems unter Einschluss heuristischer Verfahren zurück, die zeigen, wie man z. B. unter Inkaufnahme von Kompromissen hinsichtlich Lösungsqualität doch zu befriedigenden Ergebnissen gelangen kann, und bewerten diese vergleichend hinsichtlich ihrer Praxistauglichkeit.

2. Motivierende Beispiele zur Knotenüberdeckung in Graphen

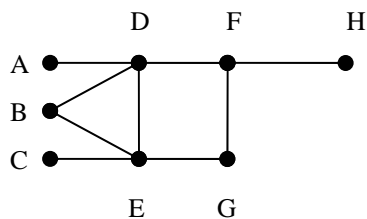
Viele Probleme aus Alltag und Wissenschaft lassen sich als Knotenüberdeckungsprobleme formulieren. Die nachfolgend genannten Fallbeispiele verdeutlichen am konkreten Problem Knotenüberdeckung, in welchen verschiedenen Bereichen das mit der P =? NP-Frage inhärent verknüpfte Phänomen der kombinatorischen 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 harmonisieren; Ziel von X ist es, möglichst viele der acht Personen so einzuladen, dass nur miteinander harmonisierende Personen an der Feier teilnehmen.

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

Eine graphische Darstellung dieses Beziehungsgeflechts sieht wie folgt aus:



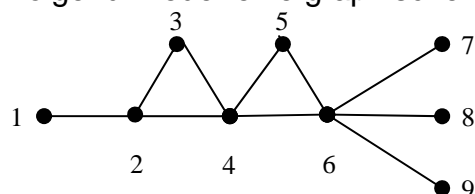
Es erweist sich, dass X mindestens drei Personen einladen darf, damit die Party harmonisch werden kann

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. So ist es in den empirischen Wissenschaften 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, so dass 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).

Nachfolgend wieder eine graphische Darstellung:



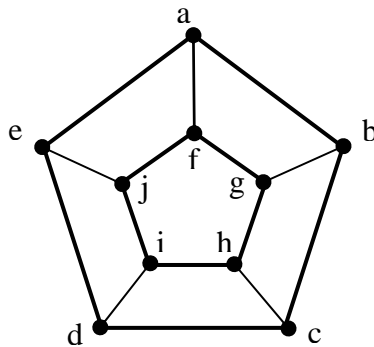
Die Elimination von drei der neun Experimente würde zu einer schlüssigen Hypothese führen.

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.

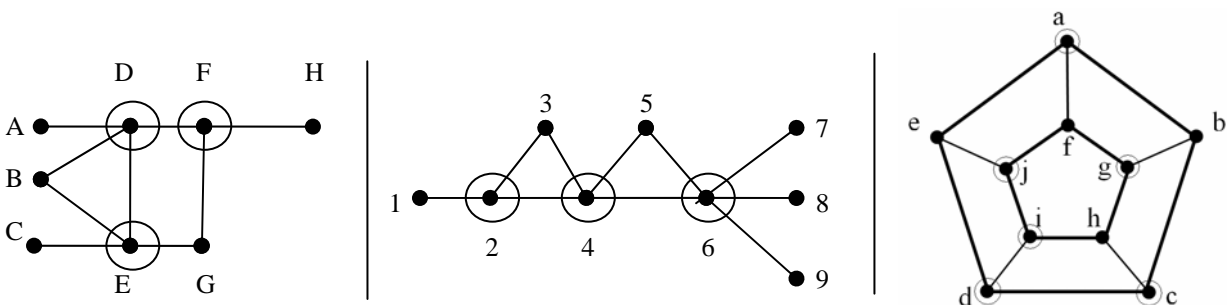
Ziel ist es, Einsicht in das gesamte Verkehrsnetz mit all seinen Straßenzügen zu haben, und dies mit möglichst wenigen Verkehrspolizisten zu erreichen. Betrachten Sie folgendes System von fünfzehn Straßen, jeweils repräsentiert durch ihre beiden „Kreuzungsendpunkte“: (a,b), (a,e), (a,f), (b,c), (b,g), (c,d), (c,h), (d,e), (d,i), (e,j), (f,g), (f,i), (g,h), (h,i), (i,j).

Auch dies läßt sich graphisch repräsentieren:



Es gibt eine Lösung mit sechs Verkehrspolizisten.

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 zugrundeliegende Problem der Knotenüberdeckung¹ (engl. Vertex Cover oder Node Cover) lässt sich also wie folgt formalisieren:

Eingabe: Ein ungerichteter Graph $G = (V, E)$.
 (V – set of vertices = Knotenmenge (Singular: vertex),
 E – set of edges = Kantenmenge)
 und eine positive natürliche Zahl k .

Aufgabe: Finde eine Teilmenge von Knoten $U \subseteq V$, so dass
 jede Kante aus E mindestens einen ihrer beiden Endknoten in U hat,
 wobei U nicht mehr als k Elemente hat

Ausgabe: Eine solche Menge U .

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

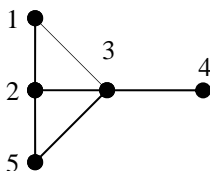
Das Knotenüberdeckungsproblem ist eines der am längsten bekannten und elementarsten Probleme aus der Menge der sogenannten NP-vollständigen Probleme. In der wissenschaftlichen Fachliteratur ist dieses Problem zusammen mit diversen Varianten vielfach untersucht worden. So findet sich auch eine Vielzahl von Anwendungsszenarien für dieses Problem, die neben den oben genannten von der Bioinformatik über Netzwerküberwachung bis hin zu „Denial of Service“-Attacken reichen (s. Glossar).

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

Wir weisen darauf hin, dass das Knotenüberdeckungsproblem auch als Optimierungsproblem formuliert werden kann: Gegeben ist ein Graph $G = (V, E)$. Finde eine kleinste Teilmenge von Knoten $U \subseteq V$, die eine Überdeckungsmenge ist.

3. Modellierungsaspekte und Problemverwandtschaft

Das Knotenüberdeckungsproblem wurde als **Graphenproblem** 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.



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

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.

Betrachtet man das Knotenüberdeckungsproblem in seiner „Matrixform“, so geht es darum, dass möglichst wenige Indizes so markiert werden, dass alle Einsen in der Matrix von den markierten Zeilen und Spalten überdeckt werden. Im Beispiel überdecken die Zeilen und

¹ Das Problem hätte man genauso gut Kantenüberdeckung nennen können.

Spalten mit den Indizes 2 und 3 alle Einsen. D. h., die Kanten von den Knoten 2 und 3 weisen zu allen anderen Knoten des Graphen.

Eine weitere äquivalente Formulierung des Knotenüberdeckungsproblems ist durch die Darstellung als **Mengenproblem** gegeben, wobei die Kanten als zweielementige 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 zweielementige 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 zweielementigen 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

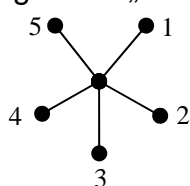
Ein Kreuz stellt also das Vorkommen eines Elements in einer Menge dar. Eine gesuchte Lösung besteht nun aus einer kleinstmöglichen Menge von Spalten (geometrisch gesprochen: senkrechten Strichen), so dass jede Zeile in wenigstens einem Kreuz getroffen wird. En passant weisen wir auf einen wichtigen, effizient lösbaren Spezialfall hin, der in dieser Modellierung deutlich zu Tage tritt: Falls es möglich ist, die Spalten so zuvertauschen, 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“:

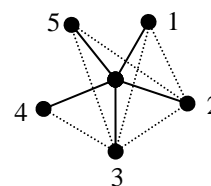
$$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_2 \vee x_5) \wedge (x_3 \vee x_4) \wedge (x_3 \vee x_5).$$

Hierbei entspricht jede aussagenlogische Variable x_i genau dem Knoten i und jede Klausel $(x_i \vee x_k)$ genau der Kante $\{i, k\}$. Ziel ist es nun, eine die Formel erfüllende Wahrheitswertbelegung der Variablen zu finden, so dass 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.

Zu guter Letzt noch eine weitere Modellierung des Knotenüberdeckungsproblems in Graphenform, jedoch als „**Kommunikations(zerstörungs)problem**“: Man betrachte folgenden „fünfeckigen Stern“:



Gestrichelt: die Verbindungskanäle

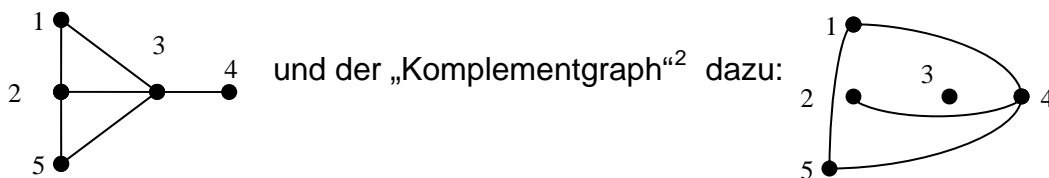


Es sollen folgende Kommunikationspaare im obigen Sternnetzwerk bestehen: (1,2), (1,3), (2,3), (2,5), (3,4), (3,5). Ziel ist es nun, möglichst wenige Kanten des Stern-

netzwerks durchzuschneiden, so dass alle Verbindungskanäle zwischen Kommunikationspaaren unterbrochen sind.

Zusammenfassend ist festzuhalten, dass wir fünf weitere äquivalente Formulierungen des Knotenüberdeckungsproblems vorgestellt haben. Insbesondere heißt das, dass es Übersetzungen der Probleme ineinander (formal gesprochen: „Reduktionen“ der Probleme aufeinander) gibt, sie also im wesentlichen immer wieder das gleiche Problem sind.

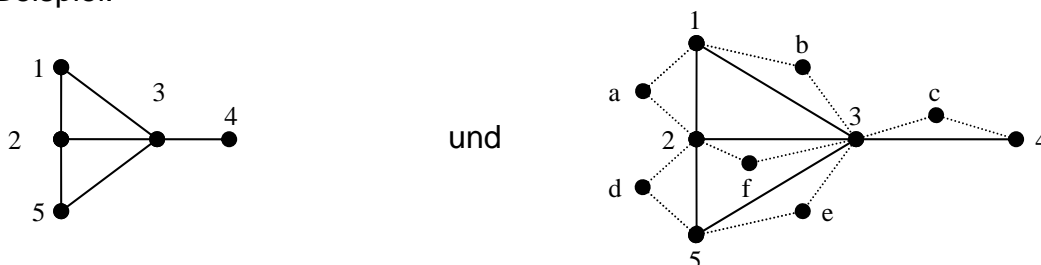
Abschließend wollen wir nun einige mit dem Knotenüberdeckungsproblem verwandte Probleme betrachten. Wir beginnen mit dem Problem der unabhängigen Mengen: Eine unabhängige Menge in einem Graph $G = (V, E)$ ist eine Knotenmenge derart, dass für jedes Paar von Knoten aus dieser Knotenmenge gilt, dass keine 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:



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“² 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. 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: Gesucht ist eine kleinstmögliche Knotenmenge, so dass jeder andere Knoten wenigstens einen Nachbarn in dieser dominierenden Menge hat. Hier ist wieder ein Beispiel:



² Der Komplementgraph entsteht aus einem gegebenen Graph, in dem man alle Kanten löscht und eine neue Kante genau zwischen solchen Knotenpaaren einfügt, wo bisher keine Kante bestand.

Der linke Graph besitzt die Knotenüberdeckung $\{2,3\}$ und die dominierende Menge $\{3\}$. Ersetzt man nun im linken Graph jede Kante durch einen Dreieckgraphen, was zum rechten Graphen mit den zusätzlichen Knoten a,b,c,d,e,f führt, so gilt folgende Aussage: der linke Graph hat eine Knotenüberdeckung der Größe k genau dann wenn der rechte 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 Rückrichtung, sprich eine Reduktion des Dominierungsproblems auf das Knotenüberdeckungsproblem zu finden.

Abschließend erfolgt noch eine ähnliche Beobachtung bei der weiter oben beschriebenen Darstellung des Knotenüberdeckungsproblems als Kommunikations(zerstörungs)problem in Sternnetzwerken. Bei letzterem handelt es sich um einen Spezialfall eines allgemeineren, ebenfalls NP-vollständigen Problems, das sogenannte „Multicut“. Erlaubt man nämlich statt Sternnetzwerken beliebige Bäume oder sogar allgemeine Graphstrukturen, so wird das Problem intuitiv noch deutlich schwerer, auch wenn es weiterhin „bloß“ NP-vollständig ist. Dies deutet an, dass innerhalb der Menge der NP-vollständigen Probleme eine reichhaltige Struktur vorliegt und im Allgemeinen die Suche nach Lösungsstrategien für jedes NP-vollständige Problem wieder neue Herausforderungen birgt. Nicht zuletzt auch hierin liegt der Reiz und die Freude an der Arbeit in diesem Gebiet.

4. Ein Verfahren zur Lösung des Knotenüberdeckungsproblems

Für die bisherigen einführenden Beispiele war die Beantwortung der Frage nach einer kleinstmöglichen Knotenüberdeckungsmenge durch einen geschulten Blick oder durch 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 gemeinsam ist, 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, eine kleinstmögliche Knotenüberdeckungsmenge zu bestimmen.

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

Dazu 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 nützt nichts: Die Antwort „nein: keine Knotenüberdeckungs Menge“ sagt nichts über die korrekte Antwort der Nachbarmengen in der gegebenen Reihenfolge! So sehr man sich auch bemüht: es wurde kein Verfahren zur präzisen Lösung des Problems gefunden, das letztlich die Durchmusterung sämtlicher Teilmengen 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 ist 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.

Wie lange benötigt ein solcher Computer, um alle Fälle durchzumustern? Die Zeit wird wie folgt abgeschätzt: da $1.000.000 = 1.000 \times 1.000 \approx 2^{10} \times 2^{10} = 2^{20}$ ist, lautet die Antwort: $2^{100} : 2^{20} = 2^{80}$ Sekunden. Da $2^{80} = 2^{10 \cdot 8} \approx 1.000^8 = 10^{24}$, entsprechen 2^{80} Sekunden etwa 10^{24} Sekunden und damit etwa 3×10^{16} Jahren. Zum Vergleich: das Universum ist ca. 13,7 Mrd. = $1,37 \times 10^{10}$ Jahre alt.

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 schnellere Computertechnik: Ein 1.000-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 Knotenüberdeckungs Menge 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. Ein möglicher Ausweg ist die Beschränkung auf Näherungslösungen.

5. Das Knotenüberdeckungsproblem als NP-vollständiges Problem

Die beschriebenen Beobachtungen sind typisch für sogenannte NP-schwere Probleme. Das Knotenüberdeckungsproblem ist ein solches Problem. Darüber hinaus trifft auch ein weiteres Phänomen zu, das im Folgenden dargestellt werden soll. Eine typische Eigenschaft klassischer Algorithmen ist nämlich ihre Determiniertheit: jeder Schritt legt eindeutig den Folgeschritt fest.

Wenn ein binärer Baum vollständig durchsucht werden soll, muss zunächst eine Suchstrategie festgelegt werden, z. B. Tiefensuche. Nach dieser Strategie wird der gesamte Baum in eindeutig bestimmter Weise durchlaufen. Dies erfordert für einen Baum der Tiefe n einen Aufwand, der proportional zu 2^n ist, d. h. exponentiellen Aufwand.

Nichtdeterminismus erlaubt eine kolossale Reduzierung dieses Aufwandes.

Wir beschreiben dieses Phänomen anschaulich als eine Schatzsuche. Das Ziel beim Durchwandern aller Blätter eines vollständigen Suchbaums war bei den motivierenden Beispielen in Abschnitt 2 das Finden einer minimalen Knotenüberdeckungs Menge. Dies sei unser Schatz, der in einem Blatt des Suchbaumes versteckt ist. Da über den Ort des Schatzes nichts bekannt ist, muss angenommen werden, dass dieser Schatz gleichverteilt über alle Blätter versteckt ist. Damit ist für einen Einzelsucher keine Suchstrategie bekannt, die nicht mindestens exponentiellen Aufwand erwarten lässt.

Ganz anders bei einer Suche, an der 2^n Personen beteiligt sind. Startpunkt der Suche ist die Wurzel des Baumes und man sucht in Richtung der Blätter. An jeder Verzweigung teilt sich die Gruppe immer wieder, bis die Blätter erreicht sind. Danach laufen alle zur Wurzel zurück. Die Suche ist erfolgreich, wenn eine Person den Schatz gefunden hat und ihn mit zur Wurzel bringt.

Der Effekt einer so organisierten Suche ist gewaltig: Der Aufwand der nichtdeterministischen Suche (die zuletzt beschriebene) ist (nur noch) proportional zu n . Der Preis hierfür sind die 2^n Sucher.

Dieses Phänomen lässt sich auch durch „Raten“ beschreiben: An jeder Verzweigung „rät“ man den richtigen Weg und die Suche wird erfolgreich, wenn es einen Weg zum Schatz gibt. Die Übertragung auf das Knotenüberdeckungsproblem bedeutet:

In einer ersten nichtdeterministischen „Ratephase“ wird eine Teilmenge U der Knotenmenge geraten. Der Aufwand hierfür ist proportional zu n , er ist also polynomial in n beschränkt; in einer zweiten (deterministischen) „Prüfphase“ wird verifiziert, ob U tatsächlich eine Menge ist, die die Knotenüberdeckung leistet: Wie bereits erwähnt ist der Aufwand hierfür ebenfalls polynomial beschränkt.

Zusammenfassend können wir feststellen:

Das Knotenüberdeckungsproblem erlaubt eine präzise Lösung mit polynomialen Aufwand, falls Nichtdeterminismus zugelassen ist. Dieses Phänomen ist typisch für sogenannte NP-Probleme. Das Knotenüberdeckungsproblem ist ein solches NP-Problem.

NP-Probleme, die auch NP-schwer sind, heißen NP-vollständige Probleme.

Das Knotenüberdeckungsproblem ist also ein NP-vollständiges Problem. Im folgenden Abschnitt werden wir diese Begriffe beschreiben.

6. NP-vollständige Mengen und das P =? NP-Problem

Das P =? NP-Problem ist ein offenes Problem der Theoretischen Informatik und der Mathematik. Die P =? NP-Frage wurde erstmals 1971 von Stephen Cook (USA bzw. Kanada) und im selben Jahr von Leonid Levin (damalige UdSSR) gestellt. Es wurde vom Clay Mathematics Institute in die Liste der Millennium-Probleme aufgenommen, für deren Lösung jeweils eine Million US-Dollar ausgelobt wurden. Das P =? NP-Problem beinhaltet die Frage nach der Inklusionsbeziehung zwischen den Komplexitätsklassen P und NP. Wie nachfolgend erläutert, gilt $P \subseteq NP$, unklar ist jedoch, ob $P = NP$ oder $P \subset NP$ gilt.

NP bezeichnet die Klasse aller Probleme, bei denen es mit einem deterministischen Algorithmus in Polynomialzeit möglich ist, zu verifizieren, ob eine geratene Lösung tatsächlich eine Lösung darstellt. Der Begriff des Algorithmus kann hierbei durch das Konzept der Turingmaschine formalisiert werden. Entsprechende Turingmaschinen arbeiten üblicherweise in zwei Phasen: in der ersten Phase wird nichtdeterministisch eine Lösung geraten, in der zweiten Phase wird diese geratene Lösung deterministisch verifiziert.

Es gibt eine zweite äquivalente Definition von NP als die Menge aller Entscheidungsprobleme, die von nichtdeterministischen Turingmaschinen in Polynomialzeit akzeptiert werden. NP enthält also alle Probleme, die von nichtdeterministischen Algorithmen mit polynomialen Zeitaufwand gelöst werden können.

Die Komplexitätsklasse P dagegen enthält alle Probleme, die von klassischen Algorithmen mit polynomialen Zeitaufwand gelöst werden können. Formal ausgedrückt umfasst die Klasse P alle Entscheidungsprobleme, welche von deterministischen Turingmaschinen in Polynomialzeit entschieden werden können. Aus diesen beiden Definitionen geht hervor, dass P eine Teilmenge von NP ist, denn deterministische Turingmaschinen sind auch nichtdeterministische Turingmaschinen, denen die Zufälligkeit, also das Raten, fehlt.

Der Begriff der NP-Vollständigkeit stammt von Stephen Cook aus dem Jahr 1971. Intuitiv ist ein Problem NP-vollständig, wenn es erstens zur Klasse NP gehört, d. h. wenn es ein NP-Problem ist, und wenn es zweitens die Eigenschaft hat, dass sich die Schwierigkeit jedes Problems aus NP in das gegebene übersetzen lässt, d. h. formal wenn jedes NP-Problem auf das gegebene Problem polynomial reduzierbar ist. Folglich lassen sich auch alle NP-vollständigen Probleme ineinander übersetzen und jedes einzelne trägt die Komplexität der ganzen Klasse in sich. Probleme, die diese zweite Eigenschaft besitzen, heißen NP-schwer. NP-schwere Probleme sind also solche, die die volle Komplexität aller NP-Probleme in sich tragen.

Alle NP-vollständigen Probleme haben zwei Gemeinsamkeiten.

- Zum einen benötigen alle bekannten deterministischen Algorithmen, die eine exakte Lösung garantieren, Exponentialzeit, d. h. exponentiellen Rechenzeitaufwand zum Finden der korrekten Antwort.
- Zum anderen hätte man für alle NP-vollständigen Probleme Lösungsalgorithmen in Polynomialzeit, falls es gelänge, für nur ein einziges Problem einen derartigen deterministischen Algorithmus zu finden. Dies hätte die Gleichheit $P = NP$ zur Folge.

Diese Gleichheit ist zwar prinzipiell möglich, aber von vielen Experten wird sie eher als unwahrscheinlich angesehen.

Eine Umfrage im Jahre 2002 unter 100 namhaften Theoretischen Informatikern zur P =? NP-Frage lieferte folgende Ergebnisse: lediglich fünf der Befragten äußerten

die Auffassung, dass diese Frage bis zum Jahr 2009 beantwortet werden kann. Immerhin 35 waren der Ansicht, dass das Problem bis zum Jahr 2049 gelöst sein wird. Sieben sagten, es wird in den Jahren zwischen 2100 und 2110 gelöst werden, weitere fünf sagten, in den Jahren zwischen 2200 und 2300. Immerhin fünf gaben an, dass es niemals gelöst werden wird.

Die Befragung zeigte weiter: 61 der Befragten haben die Überzeugung, die korrekte Antwort ist $P \subset NP$; neun gaben der Überzeugung $P = NP$ Ausdruck. Und es wird ausdrücklich erwähnt, dass es sich bei diesen neun Personen um sehr „respectable members of the community“ handelt, denen klar ist, dass sie eine Minderheitenmeinung äußern.

Der Umstand, dass 22 der Befragten das Risiko scheuten, sich festzulegen, macht deutlich, dass hinter dieser Frage mehr Unbestimmtheit steckt, als man auf den ersten Blick eingestehen will. Die Eigenschaften von NP-vollständigen Problemen führen mitunter zu der irrtümlichen Auffassung, NP sei die Klasse der Probleme, die nicht in Polynomialzeit lösbar sind. Dies ist in einem doppelten Sinn falsch:

- Aus der Definition ergibt sich unmittelbar $P \subseteq NP$, d. h. alle Probleme, die sich durch deterministische Algorithmen in Polynomialzeit lösen lassen, gehören auch zu NP.
- Außerdem gibt es Probleme, deren Lösungen sich nachweisbar nicht in polynomialer Zeit verifizieren lassen. Die gehören dann nicht zu NP: solche Probleme sind NP-schwer, aber nicht NP-vollständig.

Alle bekannten Algorithmen für NP-schwere Probleme erfordern exponentiellen Zeitaufwand und die vielfach geäußerte Vermutung lautet: diese Probleme lassen sich nicht effizient, d. h. in polynomialer Zeit lösen.

7. Lösungsstrategien

Im allgemeinen ist das Knotenüberdeckungsproblem NP-schwer und so ist nicht zu hoffen, einen im allgemeinen effizienten Lösungsalgorithmus hierfür zu finden. Dennoch bedarf dieses Problem der algorithmischen Behandlung und nachfolgend werden einige Lösungsansätze erläutert.

Ein Lösungsansatz besteht darin, die Suche nach einer optimalen Lösung aufzugeben und sich stattdessen mit einer Näherungslösung zufrieden zu geben. Solche Verfahren kennt man als (Polynomialzeit-) Approximationsalgorithmen. Für das Knotenüberdeckungsproblem ist schnell eine Lösung zu finden, die garantiert höchstens doppelt so viele Knoten enthält wie eine optimale.

Die zugrundeliegende Idee ist einfach: gemäß Definition des Problems muss jede Kante durch einen ihrer beiden Endknoten abgedeckt werden. Das Problem ist, dass im allgemeinen nicht schnell zu sehen ist, welche Wahl die günstigere ist. Der Approximationsalgorithmus wählt daher schlicht beide Endknoten der Kante. Mindestens einer der beiden Endknoten ist auf jeden Fall in einer optimalen Lösung. Wählt man nun immer wieder eine beliebige Kante, löscht ihre beiden Endknoten zusammen mit allen anliegenden Kanten bis keine Kanten im Graphen mehr übrig sind, so endet man aufgrund obiger Beobachtung mit einer Lösung, die höchstens doppelt so groß ist wie eine optimale – man spricht von einer Faktor-2-Approximation. Da der Algorithmus immer die erstbeste nichtabgedeckte Kante nehmen darf, ist leicht einzusehen, dass er in linearer Laufzeit arbeitet, sprich extrem effizient ist. Einziger aber entscheidender Wermutstropfen ist die Nichtoptimalität der Lösungsgröße.

Eine weitere effiziente, aber im allgemeinen auch nicht die optimale Lösung liefernde Strategie beruht auf folgender intuitiver Idee: Je mehr Kanten an einem Knoten anliegen, desto mehr Kanten werden bei Wahl dieses Knotens in die Knotenüberdeckung auf einmal abgedeckt. Daher liegt es nah, jeweils denjenigen Knoten zu wählen, der die aktuell meisten Kanten abdeckt, ihn und all seine anliegenden Kanten zu löschen, und zu iterieren, bis wiederum alle Graphkanten abgedeckt (sprich gelöscht) wurden. Überraschender Weise führt diese Strategie schlimmstenfalls zu einer schlechteren Approximation als die im vorigen Abschnitt beschriebene Strategie. Genauer lässt sich (nicht ganz einfach) ein Graph mit n Knoten so konstruieren, dass für ihn die derart konstruierte Lösungsmenge die $(\ln n)$ -fache Knotenanzahl im Vergleich mit einer optimalen Lösung besitzt. Jedoch ist zu vermuten, dass dieser Effekt eher nur in sehr speziellen Fällen auftritt. So erklärt sich, dass diese Strategie trotz ihres deutlich schlechteren worst case-Verhältnis hinsichtlich Approximationsgüte in praktischen Anwendungsfällen meist bessere Resultate liefert als die Faktor-2-Approximation.

Zuletzt sei noch eine in jüngster Zeit in der Forschung verstärkt untersuchte Lösungsmöglichkeit betrachtet: Der sogenannte parametrisierte Ansatz. Nehmen wir an, wir wollen untersuchen, ob ein gegebener Graph mit n Knoten eine Knotenüberdeckungsmenge bestehend aus höchstens k Knoten besitzt. Wir können nun Gebrauch machen von Beobachtungen sowohl bezüglich der oben genannten Faktor-2-Approximierbarkeit als auch der daraufgenannten „Knotengrad-Heuristik“. Der Ansatz beruht auf sogenannten tiefbeschränkten Suchbäumen. Betrachten wir zunächst wieder eine beliebige Kante $e=(u,v)$ des Graphen.

Wie bereits erwähnt, muss wenigstens einer der beiden Knoten u oder v in der gesuchten Knotenüberdeckungsmenge U , deren Größe höchstens k ist, sein. Wir machen die Fallunterscheidung: $u \in U$ oder $v \in U$. Man beachte hierbei, dass die beiden Fälle nicht notwendig disjunkt sind, sprich in beiden Fällen $u \in U$ und $v \in U$ möglich ist. In jedem Fall wird also ein Knoten gewählt und wiederum zusammen mit seinen anliegenden Kanten gelöscht. Dies wiederholen wir, bis entweder alle Kanten abgedeckt sind oder aber mehr als k Knoten für U ausgewählt wurden. Wir erhalten somit einen binären Suchbaum der Größe proportional zu 2^k , wobei jedes seiner Blätter einer derartig konstruierten Menge U (die anfangs leer gesetzt wird) entspricht. Im Gegensatz zum oben erwähnten Algorithmus des vollständigen Durchmusterens, der im wesentlichen 2^n Möglichkeiten durchprobiert, werden jetzt nur noch im wesentlichen 2^k Möglichkeiten geprüft. Dies ist von Vorteil, wenn k klein ist. Kurz gesagt: Je kleiner die Lösungsgröße ist, desto schneller lässt sich die entsprechende Lösungsmenge auch ermitteln, man spricht hier von einem parametrisierten Algorithmus, da die kombinatorische Explosion auf den Problemparameter k beschränkt werden kann. Die Suchbaumgröße lässt sich im übrigen leicht deutlich verringern: Ist z. B. ein Knoten mit nur einer anliegenden Kante vorhanden, so nehme man auf jeden Fall besser seinen Nachbarknoten (diese Wahl ist in keinem Fall schlechter), die Lösungssuche verzweigt hier also nicht in mehrere Fälle.

Sind derart alle Knoten vom Grad eins eliminiert, so hat jeder Knoten dann mindestens zwei Nachbarn (d. h. Grad zwei). Nun greift die folgende, an der „Knotengrad-Heuristik“ orientierte Strategie: Wähle einen Knoten von größtmöglichem Grad.

Es gilt: Entweder er oder alle seine Nachbarn müssen in der Lösungsmenge sein. Gibt es einen Knoten vom Grad drei oder größer, so führt dies zu einer Suchbaumgröße, die nur noch proportional zu 1.47^k ist. Man beachte dabei, dass die Suchbaumverzweigungsstrategie abgebrochen werden kann, sobald mehr als k Knoten verbraucht werden oder nur noch Knoten vom Grad höchstens zwei vorhanden sind.

In einer solchen „Grad-2-Instanz“ ist das Knotenüberdeckungsproblem wiederum effizient unter Vermeidung kombinatorischer Explosionen in Linearzeit zu lösen (eine gute Übung ist die Beantwortung der Frage „Warum?“).

Zusammenfassend lässt sich festhalten, dass das Knotenüberdeckungsproblem eine Vielzahl möglicher Lösungsstrategien besitzt, die entweder näherungsweise Lösungen in polynomialer Laufzeit oder optimale Lösungen in exponentieller Laufzeit liefern. All diese Verfahren sind leicht zu verstehen und z. T. sogar spielerisch zu erfassen. Somit können hier auf einfache Weise zentrale Themen der Algorithmik – approximative, exakte und heuristische Algorithmen für NP-vollständige Probleme – anschaulich vermittelt werden.

8. NP-schwere Probleme am Fallbeispiel des Thüringer Lehrplans

Im Lehrplan des Thüringer Kultusministeriums für das Fach Informatik an Gymnasien sind Leitlinien und Themenbereiche aufgeführt, die eine Arbeit am P =? NP-Problem ermöglichen:

Leitlinie: *Problemlösen mit Informatiksystemen*

Die Schüler sollen: Problemlösungen hinsichtlich ihrer Relevanz, Korrektheit und Effizienz beurteilen können, prinzipielle Grenzen der Formalisierbarkeit und Berechenbarkeit kennen

Leitlinie: *Auswirkungen der Informatik auf Individuum und Gesellschaft*

Die Schüler sollen einen Einblick in die historische Entwicklung von Informatiksystemen erhalten, gesellschaftlich bedeutsame Anwendungen der Informations- und Kommunikationstechniken an Beispielen darstellen können

Themenbereich 6: *Möglichkeiten und Grenzen des Einsatzes von Informatiksystemen*

Lernziele/Inhalte

theoretischer Aspekt: Ist alles mit einem Computer berechenbar? Charakterisieren des Halteproblems in PASCAL oder OBERON (Plausibilitätsbetrachtung)

praktischer Aspekt: Ist jedes prinzipiell lösbare Problem in praktisch akzeptabler Zeit auf einem Computer bearbeitbar?

Im 12. Schuljahr des Thüringer Gymnasiums sind die Schüler für Rechenzeitbetrachtungen von Problemen sensibilisiert. Sie haben diesbezüglich Erfahrungen z. B. mit dem n -Damen-Problem, mit Permutationen und mit dem Sortieren (Verfahren mit maximal quadratischer Laufzeit) gemacht.

Die Möglichkeit, die P =? NP-Frage in den Unterricht einzubeziehen, besteht also im Setzen eines Schwerpunktes darauf und in der Umordnung des bisher auf verschiedene Gebiete verteilten Stoffes. Die genannten Leitlinien des Lehrplans werden dann im Sinne der P =? NP-Frage interpretiert.

9. Weiterführende Literatur

1. M.R. Garey, D.S. Johnson:
Computers and Intractability: A Guide to the Theory of NP-Completeness;
H. Freeman 1979
Dieses Buch ist das Standardwerk der Theorie der NP-vollständigen Probleme.
2. Christos H. Papadimitriou:
Computational Complexity ;
Addison-Wesley 1994
3. J.E. Hopcroft, R. Motwani, J.D. Ullmann:
Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie;
Pearson Studium 2002
4. Ingo Wegener:
Komplexitätstheorie;
Springer-Verlag Berlin Heidelberg 2003
Diese drei Lehrbücher führen in vorbildlicher Weise ein in die Grundlagen der Berechenbarkeit und Komplexität.
5. Steven S. Skiena:
The Algorithm Design Manual;
Springer 1998
6. Th. H. Cormen, C.E. Leiserson, R. Rivest, C. Stein
Algorithmen – eine Einführung;
Oldenbourg 2004
Diese beiden Lehrbücher sind bestens geeignet für einen umfassenden Einstieg in die Algorithmik.
7. William I. Gasarch:
The $P = ?NP$ Poll;
SIGACT News 2002, Complexity Theory Column 36
Es wird eine Umfrage aus dem Jahr 2002 unter 100 namhaften Theoretischen Informatikern präsentiert, die zur weiteren Entwicklung des $P = ? NP$ -Problems befragt werden.
8. Kerner, Immo O.:
Harte Nüsse. Die schwierigsten Probleme für den Computer.
LOG IN 11 (1991) Heft 3, S. 9-17
In diesem Aufsatz werden die Mengen P und NP definiert, für beide Mengen werden Beispiele angegeben und es werden Beziehungen zwischen P und NP diskutiert.
9. Breier, Norbert:
Grenzen des Computereinsatzes. Teile 1 und 2.
LOG IN 10 (1990) Heft 1, S. 43-50; Heft 3, S. 37-42
Es werden didaktisch-methodische Überlegungen zum Thematisieren des Zeitverhaltens von Algorithmen im Informatikunterricht vorgestellt. Zusätzlich werden zwei Probleme vorgestellt und eingeordnet, die algorithmisch unlösbar sind.
10. Thüringer Kultusministerium:
Lehrplan für das Gymnasium. Informatik
Erfurt, 1999;
Im Lehrplan des Thüringer Kultusministeriums für das Fach Informatik an Gymnasien sind Leitlinien und Themenbereiche aufgeführt, die eine Arbeit am $P = ? NP$ -Problem ermöglichen

Glossar

Adjazenz (lat. *adiacere*, angrenzen) Verbundenheit; wird bei Graphen verwendet, bei denen Knoten mittels einer Kante verbunden (adjazent) sind

Adjazenzmatrix Tabelle, die die Adjazenz von Knoten anzeigt.

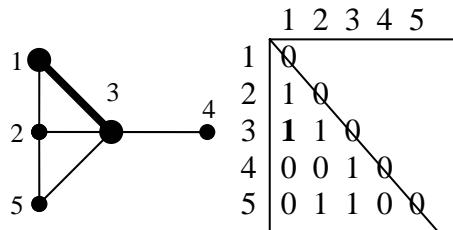


Abb: Adjazenz wird mit einer „1“ ausgedrückt, ansonsten steht eine „0“

„Denial of Service“-Angriffe Ein Server wird mit einer Flut von Anfragen überhäuft und kann auf diese Art lahm gelegt werden

Graph Geordnetes Paar (V,E) aus einer Menge von Knoten, engl. *vertex* und einer Menge von Kanten, engl. *edge*. Die Knoten heißen im Text u, v , die Kanten e .

kombinatorische Explosion Die Zahl aller Möglichkeiten zum Durchprobieren wächst exponentiell, s. auch Abschnitt 4

konjunktive Normalform, 2- konjunktive Normalform Form einer aussagenlogischen Verknüpfung, bei der Disjunktionsterme (oder-verknüpfte Terme) konjunktiv verknüpft sind. Bei der im Artikel verwendeten 2-konjunktiven Normalform bestehen die Disjunktionen immer aus 2 Literalen.

Nichtdeterminismus Zu einer bestimmten Eingabe kann eine Turingmaschine mehrere Möglichkeiten für den Übergang in den nachfolgenden Zustand besitzen.

P =? NP-Frage Es gilt $P \subseteq NP$. Offen ist die Frage der Mengengleichheit.

Polynomiale Reduktion Jede Instanz des einen Problems wird eindeutig auf eine Instanz eines anderen Problems abgebildet, wobei die Abbildung von einer Turingmaschine in Polynomialzeit zu berechnen ist

Polynomial beschränkter Aufwand Der Aufwand wird durch eine Funktion beschrieben, die durch ein Polynom beschränkt ist