PROF. DR. SÁNDOR FEKETE

Abteilung Algorithmik, TU Braunschweig

THEMA

GEHT'S NICHT NOCH ETWAS SCHNELLER?

Eine Rundreise durch gelöste und ungelöste Probleme des algorithmischen Alltags

Algorithmen

Datenstrukturen

I get the job done. What the hell do you want?

Can you make it without killing yourself?

I don‘t make things difficult. That‘s the way they get, all by the

Ordentlich, hält Regeln ein

work with you.

Locker drauf, zielorientiert

Ain‘t got no choice.

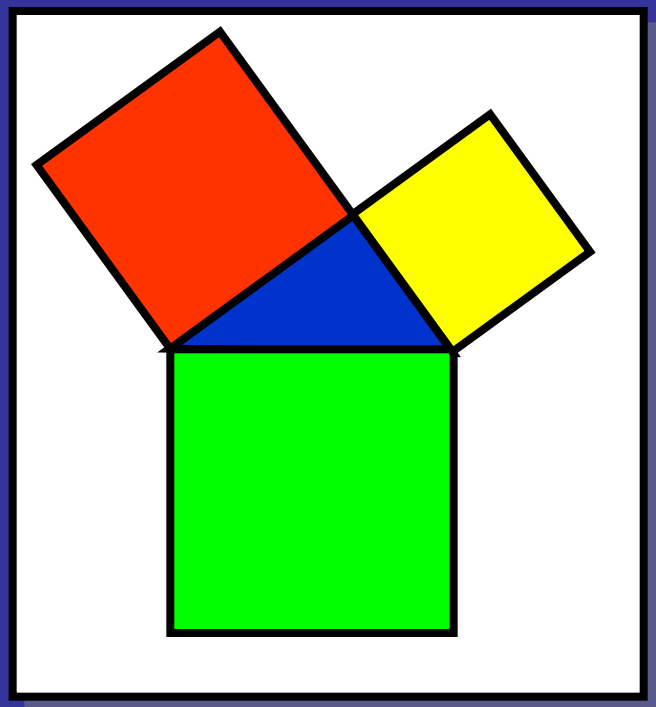
Then don‘t.

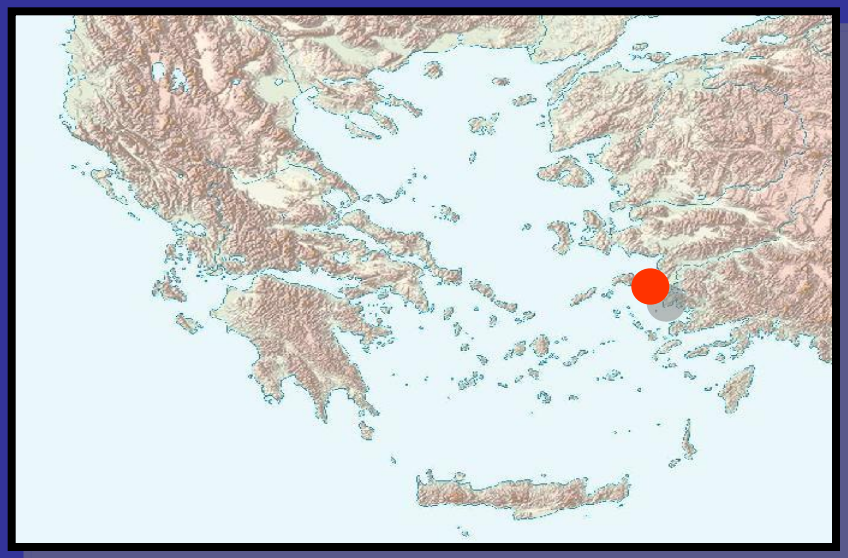
I‘m getting too old for this…

mathematische

Die bekannteste Formel aller Zeiten

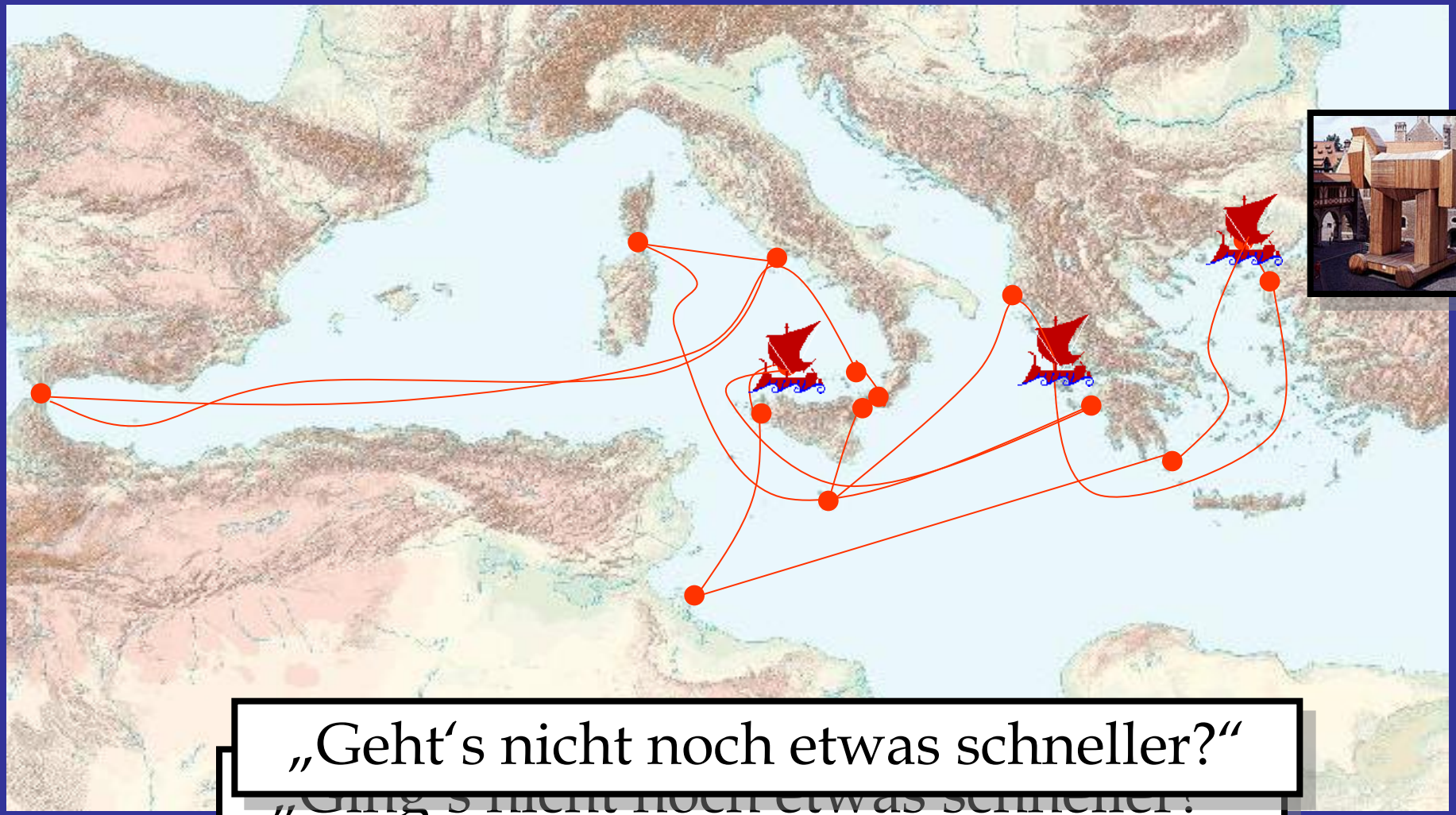
$$a^2 + b^2 = c^2$$





Pythagoras (ca. 500 v.Chr.)

Ein 20 Jahre früher

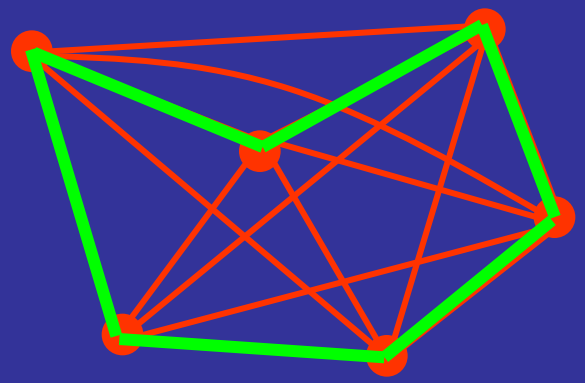


„Geht‘s nicht noch etwas schneller?“

„Ging‘s nicht noch etwas schneller?“

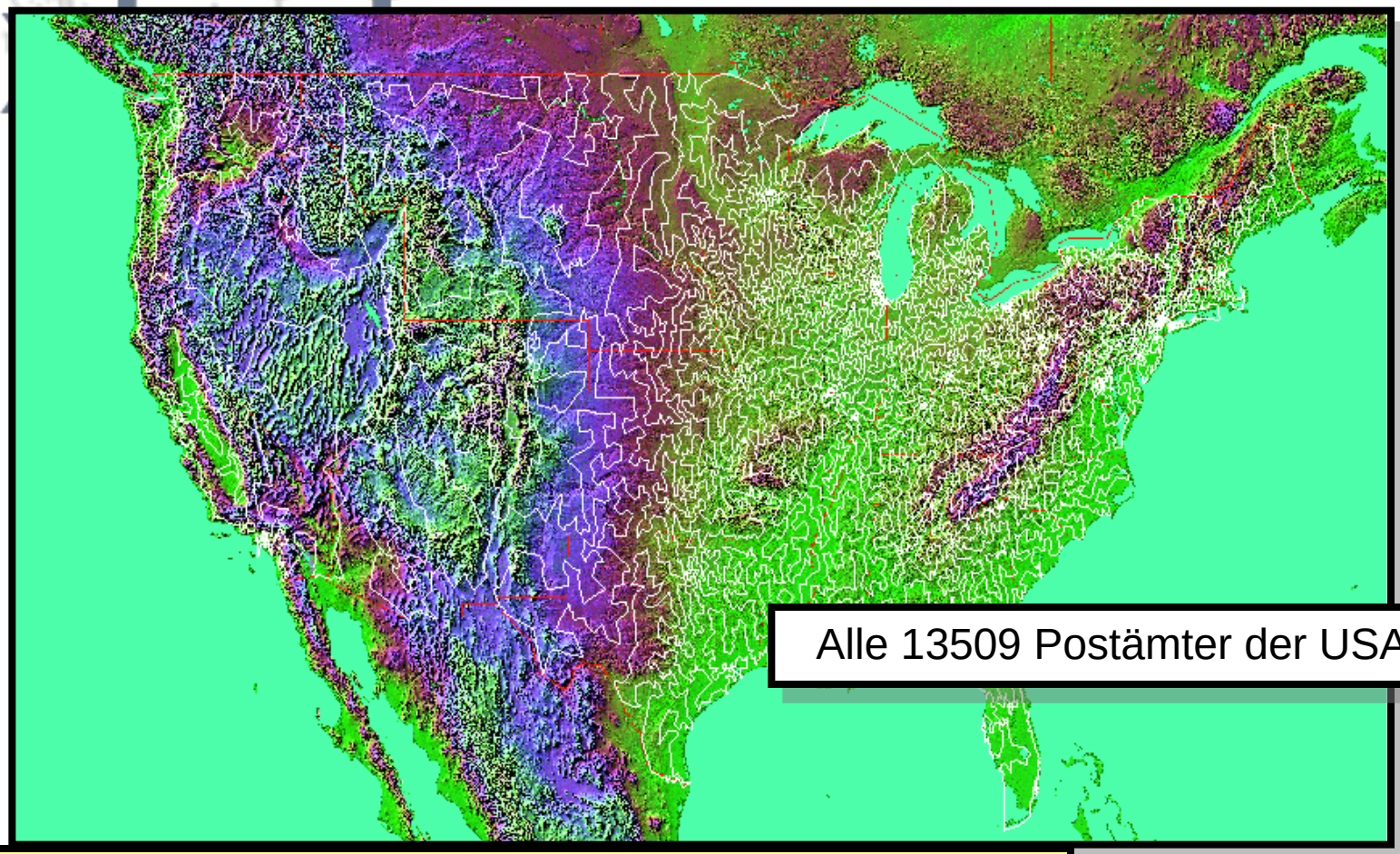
Das Rundreiseproblem

Gegeben: Ein Graph $G = (V, E)$
mit Kantenlängen w_e



Gesucht: Eine kürzeste Rundreise

Eine optimale Rundreise in der Neuen Welt



Alle 13509 Postämter der USA

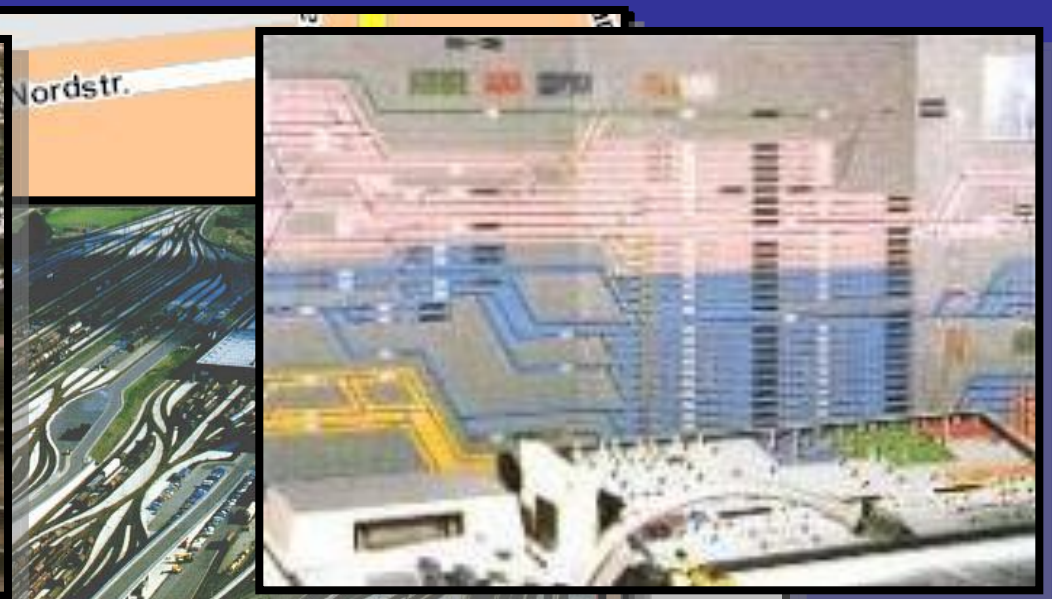
What‘s the complexity class of the best linear programming cutting-plane techniques? I couldn‘t find it anywhere. Man, the Garfield guy doesn‘t have

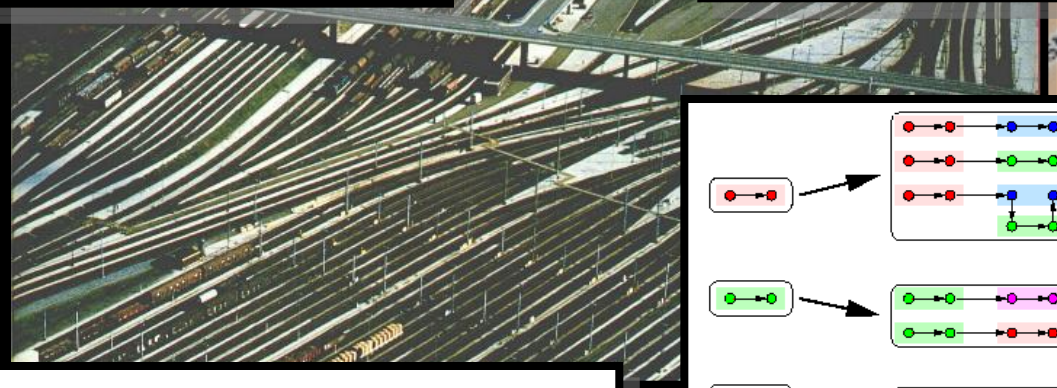
10 Jahre (!) CPU-Zeit

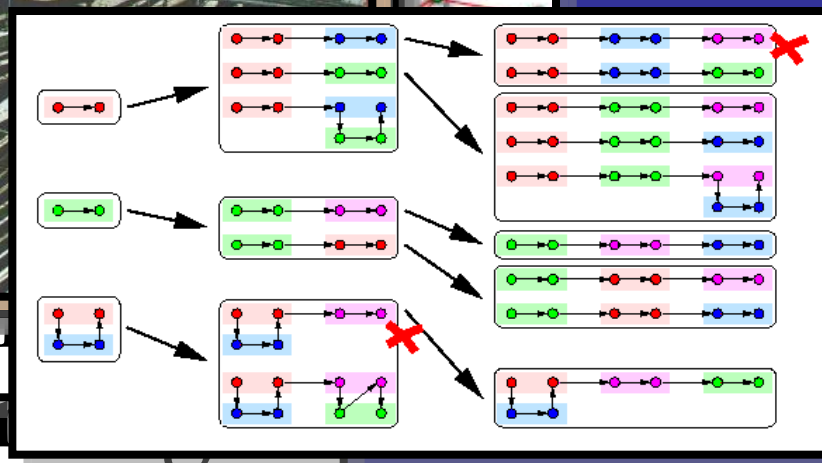
Beale-Orchard-Hayes-Preis 2000

schneller?“

Probleme optimaler Reisegestaltung







Eisenbahnplanung

Andere diskrete Optimierungsprobleme

Stabile Kommunikationsnetzwerke

Ein algorithmisches Problem

Gegeben: n Puzzleteile

Einfach so:

$O(n^2)$

Für n=5000: 12.502.500

Raffiniert sortiert:

$O(n \log n)$

Gesucht: Eine „richtige“ Anordnung der Teile

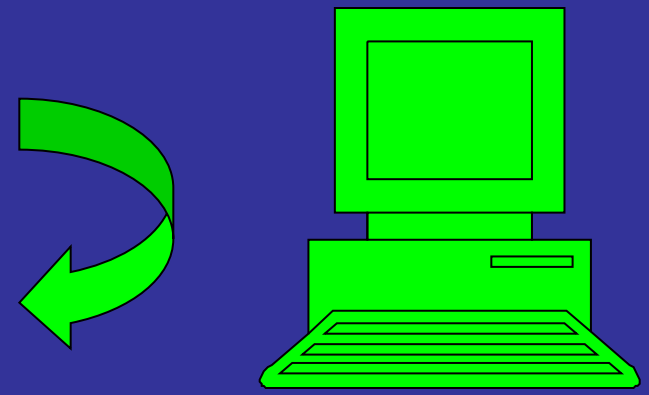
Eine systematische Methode zum Puzzeln

Konstruktives Lösen von Problemen

Gegeben: Eine Problembeschreibung

Gegeben: Eingabedaten

Gesucht: Ausgabedaten



Gesucht: Ein Algorithmus, der das Problem löst

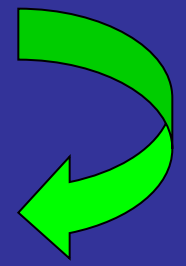
„Geht‘s nicht noch etwas schneller?“

Die Laufzeit von Algorithmen

Gegeben: Eine Problembeschreibung der Größe O(n)

Gegeben: O(n) Zahlen

Gesucht: O(n) Zahlen



$O(n^k)$

Die Klasse P

Gesucht: Ein polynomieller Algorithmus

Ein polynomieller Algorithmus

Sortieren von Objekten

Gegeben: n Objekte unterschiedlicher Größe

23 17 13 19 33 28 15

Gesucht: Eine Sortierung nach Größe

Zahl der Vergleiche: $O(n \log n)$

„Geht‘s nicht noch etwas schneller?“

Ein untere Laufzeitschranke

Gegeben: n Objekte in einer beliebigen Anordnung

(1) Insgesamt gibt es n*(n-1)*…*2*1 = n! mögliche Anordungen

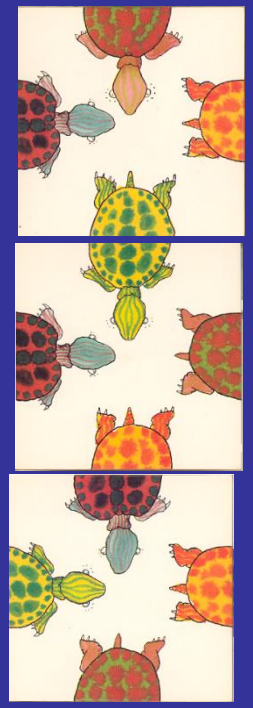
(2) Jeder Vergleich teilt die verbleibenden Anordnungen in zwei Mengen

(3) Im schlechtesten Fall bleibt die größere Menge übrig

(4) Insgesamt benötigen wir $\log(n!) = \Omega(n \log n)$ Vergleiche.

Ein algorithmisches Problem

Gegeben: n Puzzleteile



Probieren:

$$\Omega(n!\,4^{n-1})$$

Für n=9: 23.781.703.680

Raffiniert?!

Problem: „Sackgassen“!

Gesucht: Eine „systematische“ Methode zum Legen von Puzzeln

Kofferpacken für die Reise

Gegeben: Eine Menge von n Objekten, jedes mit einer Größe l_i; Gesamtgröße $\sum_{i=1}^{n} l_i = 2K$

29 13 17 31 35

24 31 7 53

120

120

„Geht‘s nicht noch etwas schneller?“

Gesucht: Eine Verteilung auf zwei Koffer der Größe K

Eigenschaften des Problems

(1) Wenn es eine Lösung gibt, so lässt sie sich ziemlich schnell *NachPrüfen*.

Die Klasse NP

(2) Wenn es *keine* Lösung gibt, lässt sich das anscheinend nur schlecht *beweisen*.

Alle 2^{n-1} Möglichkeiten?!

Probleme, für die man pfiffig „schnell“ eine Lösung *Pfinden* kann:

Die Klasse P

(Echt pfundig - gepfundes Pfressen für Algorithmiker!)

Wo ist mein Schlüssel??

Ein komplexitätstheoretischer Glaubenssatz

Karl Popper (1902-1994)

Universalaussagen lassen sich nicht verifizieren, sondern nur falsifizieren.

Also: Nachprüfen allein ist noch lange nicht Pfindig!

(Oder Pfiffig)

Noch schlechtere Nachrichten

Ein

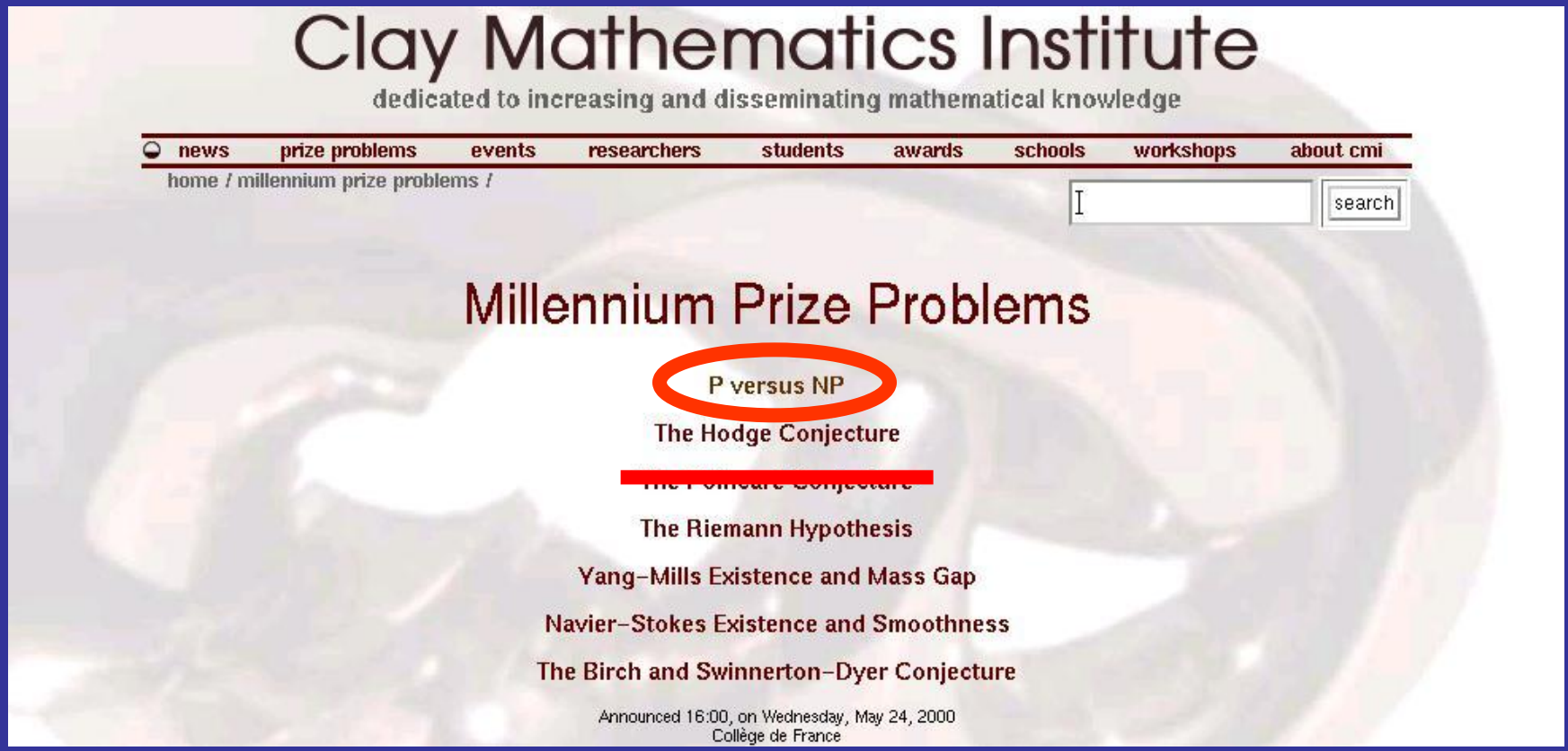
bede

Satz von Cook (1973):

Es gibt schwerste Probleme in NP, sogenannte „NP-vollständige Probleme“

$$P = NP$$

Ein Milleniumproblem



Preisgeld: 1.000.000 $

Was tun bei NP-Vollständigkeit?

Idealer Algorithmus:

Liefert

1. ~~immer~~
2. ~~schnell~~
3. ein ~~optimales~~ Ergebnis.

Strategien in schwierigen Situationen:

(A) Mit dem Schicksal hadern + diskutieren
(B) Auf Glück vertrauen
(C) Hart arbeiten
(D) Erwartungen zurückschrauben

Algorithmische Strategien:

(A) Komplexitätsanalyse: Gibt es Fälle, die nicht NP-vollständig sind?
(B) Heuristiken: Raten und hoffen
(C) Algorithmen-Tuning, um konkrete Instanzen optimal lösen zu können.
(D) „Gut“ statt „optimal“: Approximationsalgorithmen

Das bekannteste mathematische Problem für 350 Jahre

$$a^n + b^n = c^n$$

Fermatsche Vermutung (ca. 1650):

Es gibt keine ganzzahlige Lösung für n>2.

Beweis“ (Fermat)

Beweis: Wiles 1997

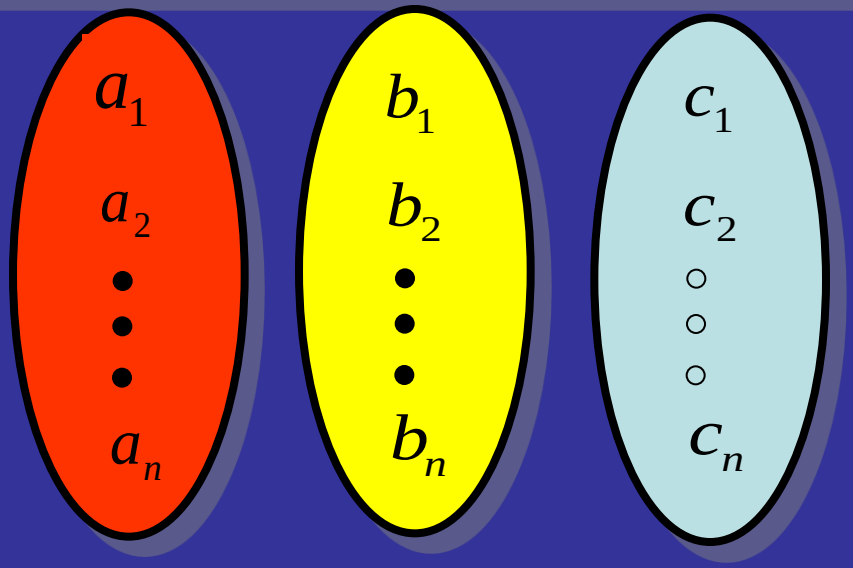
„Wahrhaft wunderbar, leider ist der Platz z

Ein ungelöstes algorithmisches Problem der Gegenwart

$$a^n + b^n = c^n$$

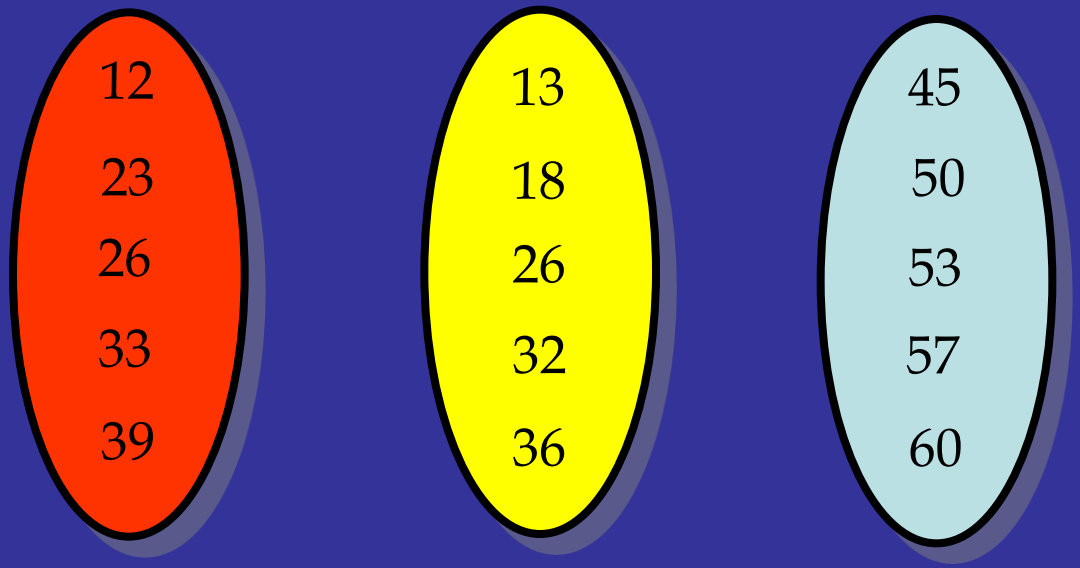
3SUM

Gegeben: Mengen A, B, C



Gesucht: i, j, k mit $a_i + b_j = c_k$

Wie schwer ist 3SUM?

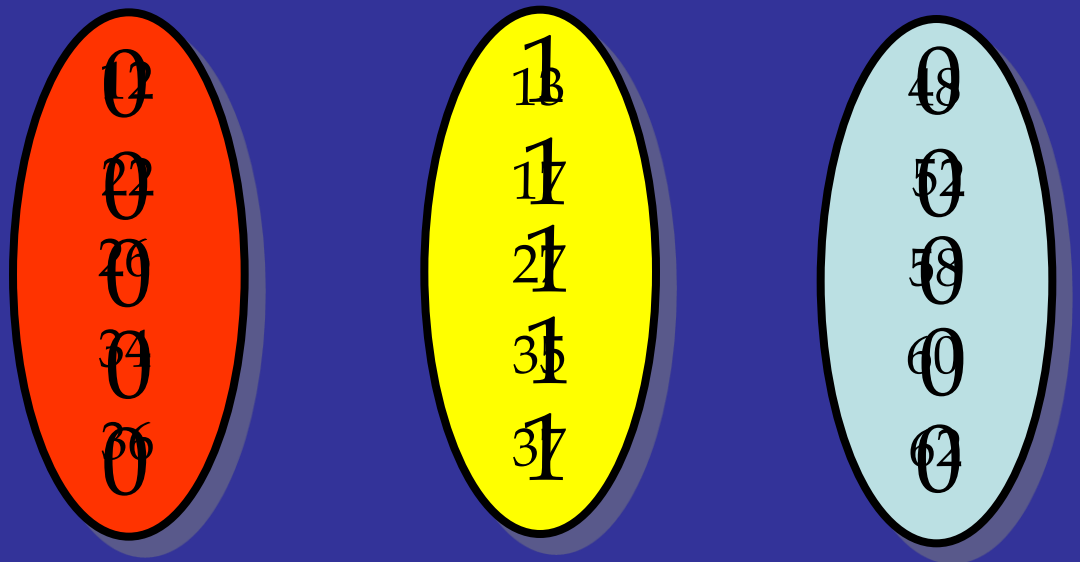


Gesucht: i, j, k mit $a_i + b_j = c_k$

Gibt einen Algorithmus schneller als $O(n^2)$?

Demaine et al. 2005: $O(n^2 \log\log n / \log n)$

Wie schwer ist 3SUM?



modulo 2

Gesucht: i, j, k mit $a_i + b_j = c_k$

Gibt‘s nicht!!

Warum ist 3SUM interessant?

Komplexitätstheorie

5SUM

bjekt mit n Kanten

Gegeben: n^2 Zahlen

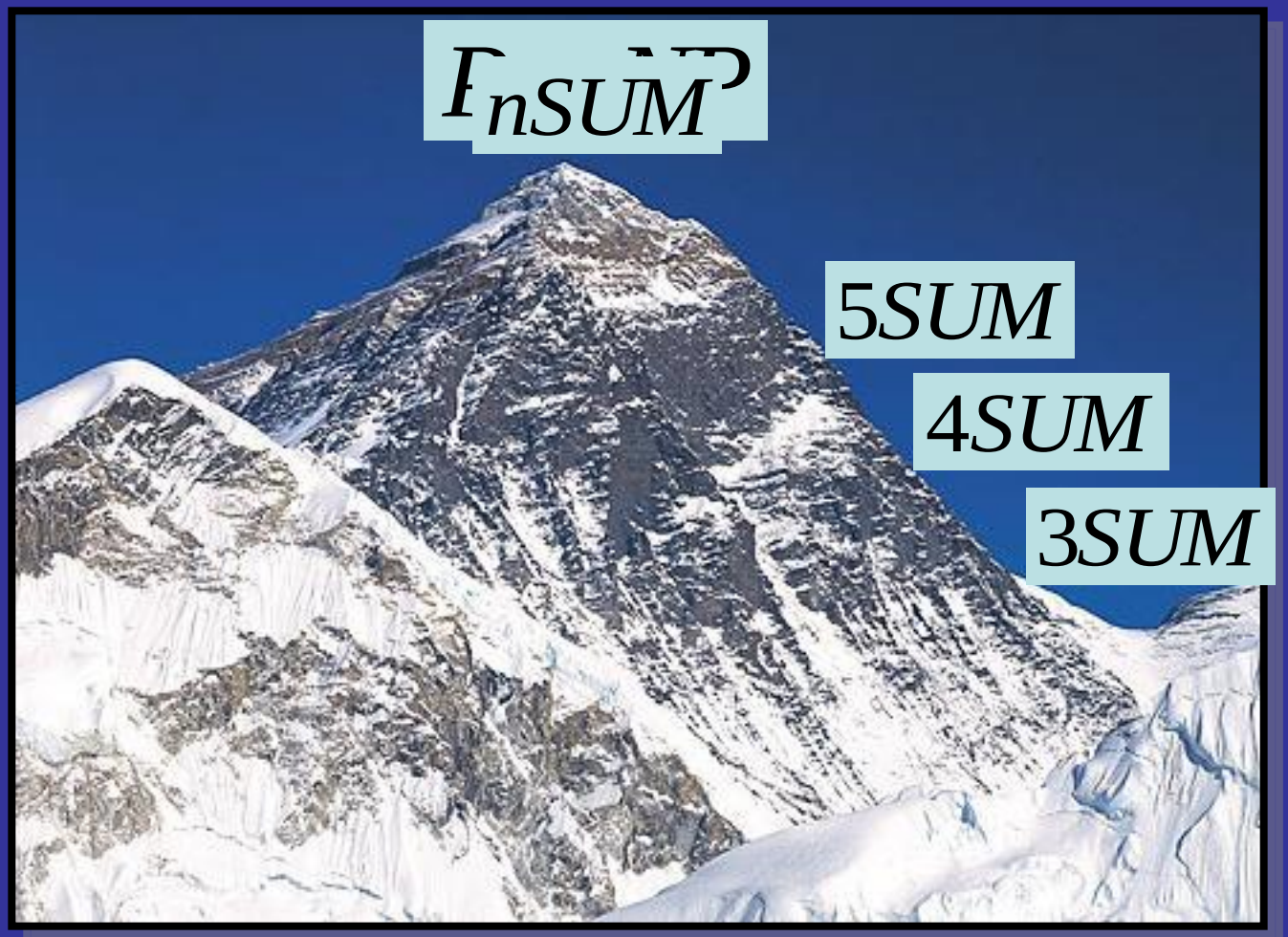
Gesucht: 4 Zahlen

$\Omega(n^{n-1})$?!

Noch ein bewegtes Objekt mit n Kanten

Kollidieren die Objekte?

Eine Idealroute zum Gipfel



Mt. Everest Südwestflanke

Mehrdimensionales Kofferpacken

BRAUNSCHWEIGER ZEITUNG
WOLFENBÜTTELER ZEITUNG UND ANZEIGER

Samstag, 28. Januar 2006 unabhängig · nicht parteigebunden Nr. 24 · 61. Jahrgang · 1,20 €

Wenn Mathematiker Autos beladen

TU-Professor Sándor Fekete tüftelt an ganz handfesten Alltagsproblemen, kombiniert und optimiert

Von Cornelia Steiner

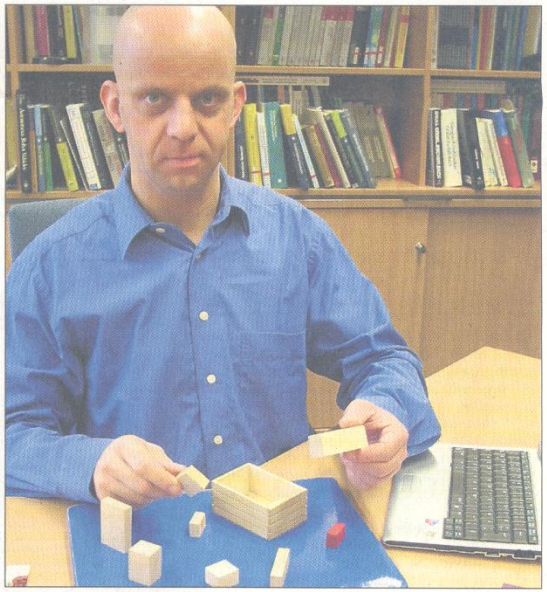
Sieben große Probleme müssen Mathematiker weltweit lösen, jeweils eine Million Dollar Preisgeld sind dafür ausgeschrieben. Bei manchen von ihnen versteht man selbst die Fragestellung erst nach einigen Fachsemestern. Das gesteht sogar Sándor Fekete, Professor für mathematische Optimierung an der TU Braunschweig. Er hat auch nicht den Anspruch, eines dieser abstrakten Probleme zu lösen. Stattdessen widmet er sich mit Kollegen und Studenten den handfesten Fragen des alltäglichen Lebens. Denn Mathematik hat eben doch etwas mit dem Hier und Jetzt zu tun: mit Routenplanung zum Beispiel, mit dem Packen eines Kofferraumes oder sogar mit Rasenmähen. All das sind Probleme, bei denen es darum geht, aus einer bestimmten Menge möglicher Kombinationen die beste Lösung zu finden.

> „Spannend wird das Lösen von Optimierungsproblemen dadurch, dass man oft nicht alle Informationen zur Verfügung hat."
>
> Sándor Fekete

In jedem Fall gilt: „Man braucht immer einen Algorithmus, der unabhängig vom Einzelfall eine systematische Lösung anbietet", sagt Fekete. Für die Routenplanung bedeutet das: Ein einziger Algorithmus liefert sowohl den kürzesten Weg von dar. In unserem Fall wären die Knoten Kreuzungen und die Verbindungen Wege. Bevor gerechnet wird, muss der Umfang des Graphen allerdings einschränkt werden: Für die Fahrt nach Peine ist schließlich das Umfeld von Helmstedt uninteressant. „Wenn so eine Struktur vorhanden ist, kann man sehr gut ermitteln, wie man am schnellsten zum Wunschort kommt", sagt er.

Doch alles hat auch seine Tücken, denn es gibt einfache und schwere Probleme: Einfach ist es, den kürzesten Weg zwischen zwei Punkten zu berechnen. Einfach ist es auch, die schnellste Verbindung in einem Fahrplan zu finden, oder die billigste Flugverbindung zu ermitteln.

Schwierig wird das Ganze aber, sobald mehrere Wegeprobleme ineinander greifen: Das ist der Fall, wenn der kürzeste Weg über mehrere Punkte in unbestimmter Reihenfolge berechnet wird. Wer also eine schnelle Rundreise durch alle Braunschweiger Kneipen plant, sollte wissen, wie das funktioniert. Kompliziert ist es auch, einen guten Bahnfahrplan zu erstellen, oder Flugpläne. Denn dann müssen zusätzliche Aspekte beachtet werden – Art der Flugzeuge, Größe der Besatzung, Strecken, Anzahl der



Professor Sándor Fekete zeigt auf einfache Weise die Probleme, die sich beim Packen und Beladen eines Kofferraumes ergeben können. Fotos (2): Steiner

nicht alle wichtigen Informationen zur Verfügung hat", sagt er. So ist zum Beispiel ungewiss, ob ein Stau auftritt oder wie viele Passagiere tat-

sich mit modernen Computersystemen und großen Kommunikationsnetzen beschäftigen, aber auch mit

Paradigmenwechsel

Klassi

1. vollstä
2. ein zer
3. Verteil

unprol

1.
2.
3.

Komplexe Systeme erfordern eine Erweiterung des algorithmischen Denkens!

Alles fließt? Schön wär's. Der Stau ist eine fast tägliche Erfahrung. Wie hier auf der A2. Foto: Archiv

zur Entfaltung bietet.

Zurück zum Stau. Fekete: „Gucken Sie sich doch mal an, was die Autos heute so alles können, aber untereinander sind sie völlig unvollkommen vernetzt." Schön wäre es zum Beispiel beim Stop-and-Go-Verkehr, mehr zu wissen, was die anderen Autos davor tun.

Manche Staus scheinen wie aus einem Nichts zu entstehen. Man spricht auch von einem Phantomstau. Der Vorgang ist freilich seit Jahren enträtselt. Fekete: „Bei hohen Verkehrsdichten kann sich schon eine kleine Störung, etwa ein starkes Abbremsen oder ein zu dichtes Auffahren, nach hinten verstärken, durch den gesamten Verkehr fortpflanzen und ein ‚Schwingen' des Verkehrssystems auslösen."

Wenn sie einmal da sind, lassen sich die Stauwellen nicht mehr so einfach ausbügeln. Schon kleine Störungen genügen, um die Wellen zu erzeugen. Wie so ein „Ruckeln" entsteht, lässt sich sowohl in Simulationen wie auch in Experimenten beobachten. Fekete: „Das Auftreten von Stauwellen ist in den physikalischen Gesetzen verankert."

„Das Auftreten von Stauwellen ist in den physikalischen Gesetzen verankert."
Sándor Fekete, Verkehrsforscher

Was bringt der Versuch der TU-Forscher, Fahrzeuge mit besseren Daten über den vorausfahrenden Verkehr zu versehen? Eine ganze Menge. Bereits jetzt kann man mit drahtloser Kommunikation Informationen über Ort und Geschwindigkeit von Fahrzeugen weitergeben, und es existieren, wie Fekete erläutert, auch schon Systeme, mit denen sich ein Auto an die Geschwindigkeit eines unmittelbar vorausfahrenden Autos anpassen kann.

Die beschriebenen Stauwellen lassen sich jedoch damit nicht beseitigen. Hierfür muss die Wechselwirkung zwischen den Fahrzeugen so beeinflusst werden, dass sich auch die physikalischen Gesetzmäßigkeiten für den Fahrzeugfluss verändern.

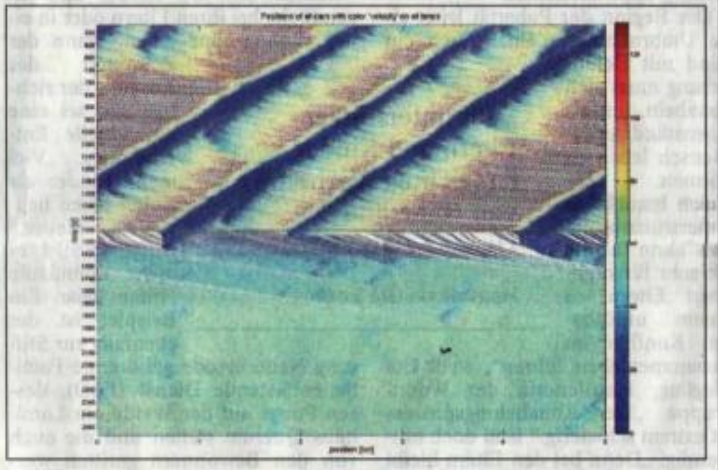
Die Arbeitsgruppe um Sándor Fekete hat nun ein einfaches Regelwerk entwickelt, mit dem Autos die Informationen mehrerer vorausfahrender Autos auswerten können, um den Verkehrsfluss im Stau zu glätten. Erkennt der Fahrer, gut informiert, dass sich der vorausfahrende Verkehr verlangsamt, muss er nicht unnötig beschleunigen.

Das hat einen verblüffenden Effekt auf den gesamten Verkehrsfluss.

Fekete: „In Simulationen zeigt sich eine Treibstoffersparnis bis zu 40 Prozent." Der Verkehr kann gleichmäßiger fließen, und der Stau kann sich sogar wieder auflösen, selbst wenn nur ein Teil der Fahrzeuge mitspielt.

Da jeder Fahrer Sprit spart, gibt es einen starken Anreiz, das System zu benutzen und die Regeln einzuhalten. Und die Nerven werden geschont. Freilich muss das ganze an der TU ersonnene System tatsächlich noch eingebaut werden. Das ginge auch bei älteren Autos. Und die Industrie muss mitspielen. Man darf sich freilich schon ein bisschen wundern, dass die großen Autokonzerne dieses Thema bisher wenig beachteten.

Der Stau und die Intelligenz, seine Schockwellen zu mäßigen. Ein Graffito an der A 40 verweist auf den eigentlichen Verursacher: „Ihr steht nicht im Stau, Ihr seid der Stau!"



Grafik „Simulation von Stau und Auflösung": Im oberen Teil der Simulation erkennt man typische Stauwellen, dargestellt als blaue Streifen. In der Bildmitte wird das neue, an der TU entwickelte Regelwerk eingeschaltet. Der Stau löst sich weitgehend auf. Bild: Sándor Fekete

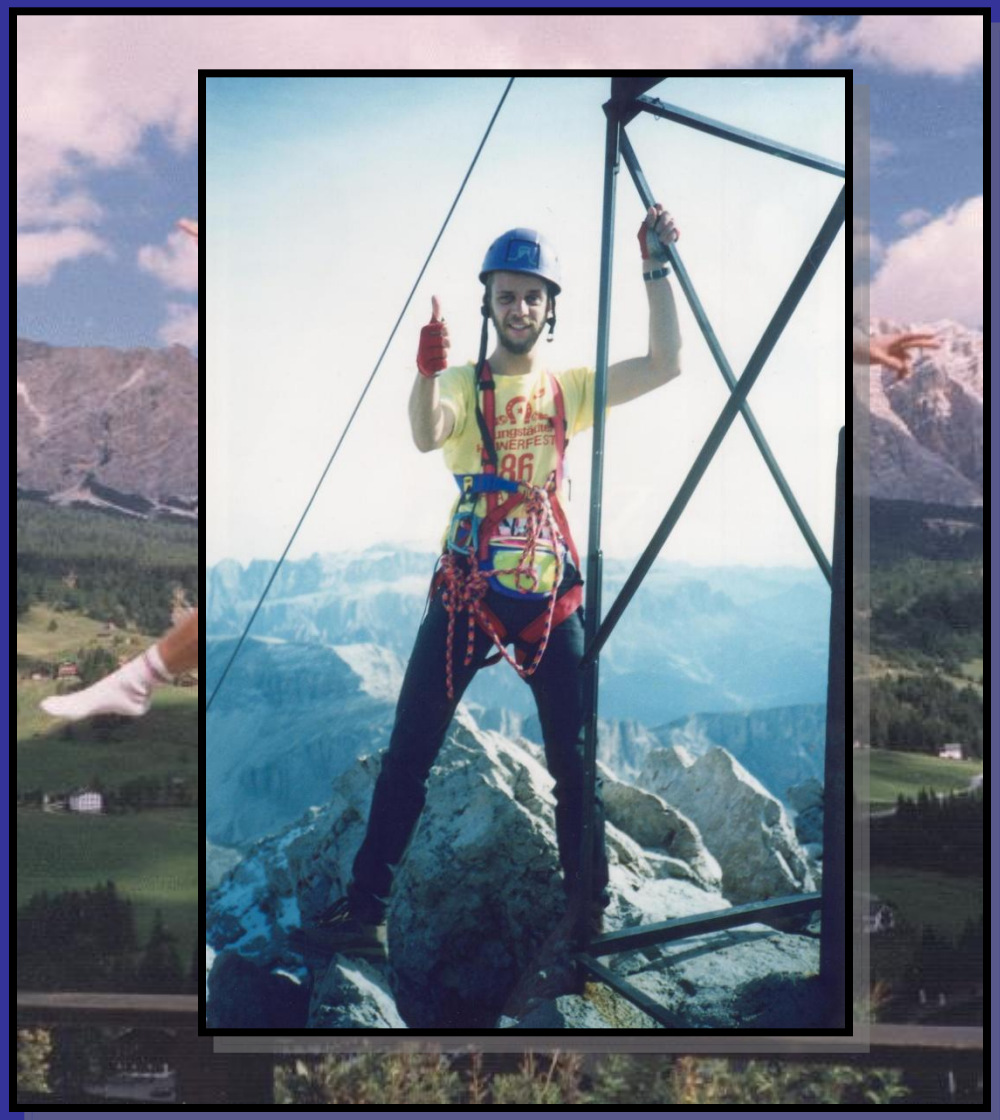
LEBENSDATEN

Professor Dr. Sándor Fekete

45 Jahre. Studium der Mathematik und Physik in Köln, Diplom 1989. Promotion (Mathematik) in Waterloo (Kanada) 1992. Forschungsaufenthalte in Stony Brook (USA), Tel Aviv, London, Newcastle (Australien), Kingston (Kanada) und am Massachusetts Institute of Technology (MIT).

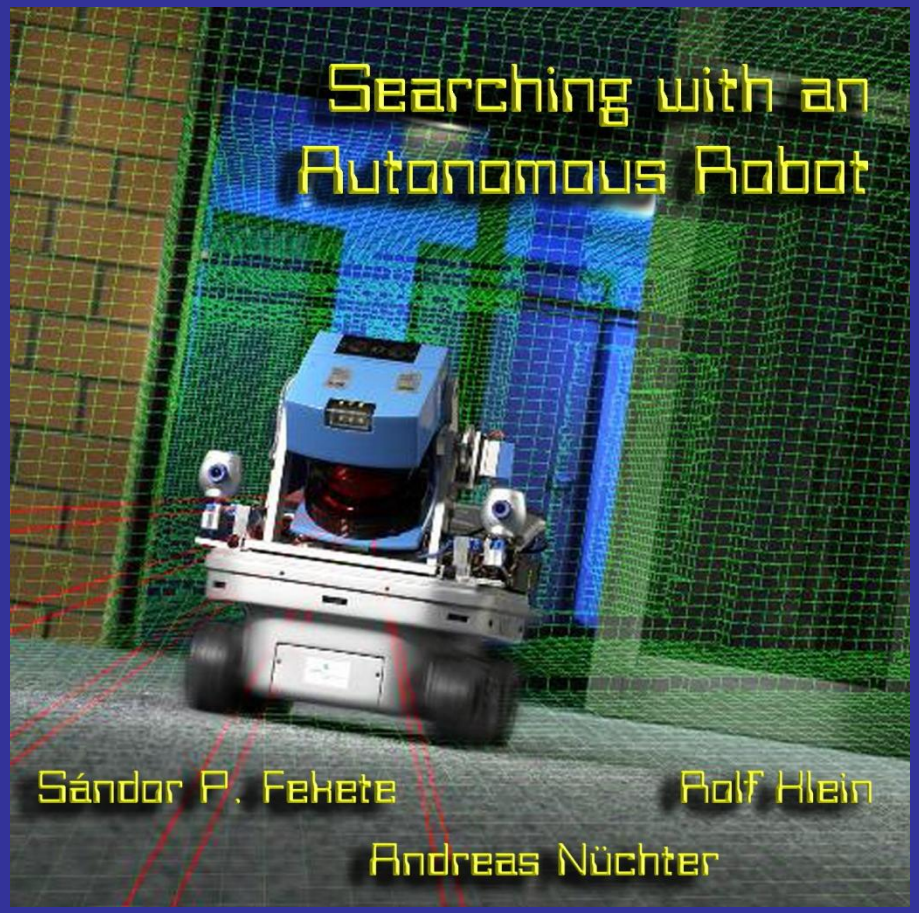
Habilitation in Köln 1998. Danach an der TU Berlin. Seit 2001 Professor an der TU Braunschweig, zunächst in der Mathematik, seit 2007 in der Informatik.

Es gibt noch viele Gipfel…



Literaturtipp:

Peter Gritzmann, Renè Brandenberg:

Das Geheimnis des kürzesten Weges

Routenplanung mit unvollständiger Information