Informatische Lösungsmethoden für schwere Berechnungs- und Entscheidungsprobleme

Georg Gottlob

Institut für Informationssysteme

Technische Universität Wien

Gliederung

- Was ist ein Entscheidungsproblem / Berechnungsproblem ?
- Komplexität von Algorithmen und Problemen
- NP-vollständige Probleme
- Drei Lösungsansätze:
  - Randomisierte lokale Suche
  - Approximation
  - Identifikation leicht lösbarer Subklassen

Übersichtsvortrag, keine mathematischen Details.

Das Rucksackproblem

GEGENSTAND	GEWICHT	WERT
Luster	5500	14300.-
Schatulle	3200	8000.-
Schwert	1500	8500.-
Bild	3400	6800.-
...	...	...

Allgemeine Problemformulierung:

Angabe: Gegeben Liste von Einträgen $\langle Gegenstand, Gewicht, Preis \rangle$, sowie Maximales Gewicht G, gewünschter Wert W

Frage: Gibt es eine Menge $S \subseteq$ Gegenstände, sodaß

$$\sum_{x \in S} gewicht(x) \leq G \quad \text{und} \quad \sum_{x \in S} preis(x) \geq W \quad ?$$

Berechnungsproblem

Berechne eine Lösung S, so daß

$$\sum_{x \in S} gewicht(x) \leq G \quad \text{und} \quad \sum_{x \in S} preis(x) \geq W$$

Berechnungsproblem mit Optimierung

Berechne eine Lösung S, so daß

$$\sum_{x \in S} gewicht(x) \leq G \quad \text{und} \quad \sum_{x \in S} preis(x) \text{ maximal}$$

Ähnliche Probleme: Frachtraumplanung, Lagerplanung,
Verschnittoptimierung, Anlagenoptimierung.

Problem/Probleminstanz

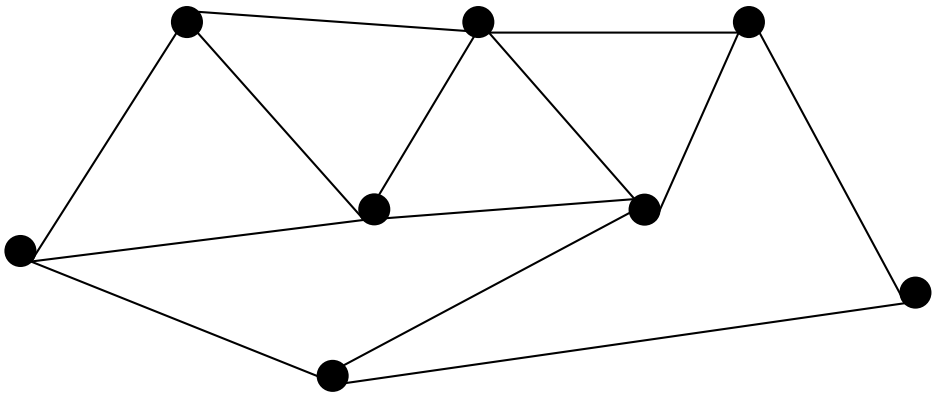
Problem: Abstrakte Problemstellung ohne konkrete Daten.

Instanz: Konkrete Angabe.

Zu einem Problem existieren i.a. unendlich viele Instanzen.

Größe einer Instanz: Anzahl der verwendeten Zeichen.

Viele interessante Probleme beziehen sich auf Graphen:

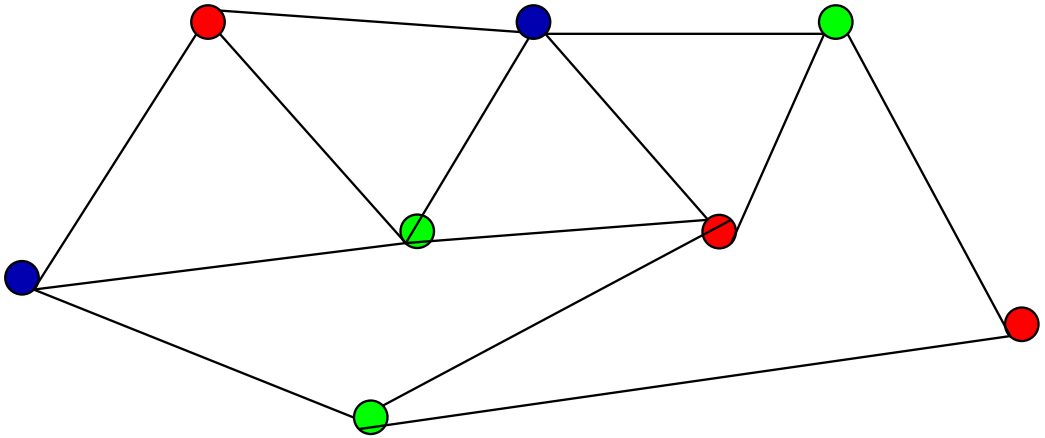


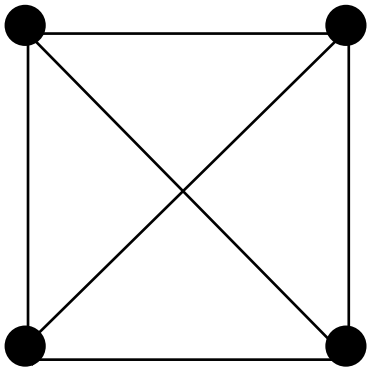
Graph-Dreifärbbarkeit

Angabe: Ein Graph G.

Frage: Ist G dreifärbbar?

Beispiele für Instanzen:



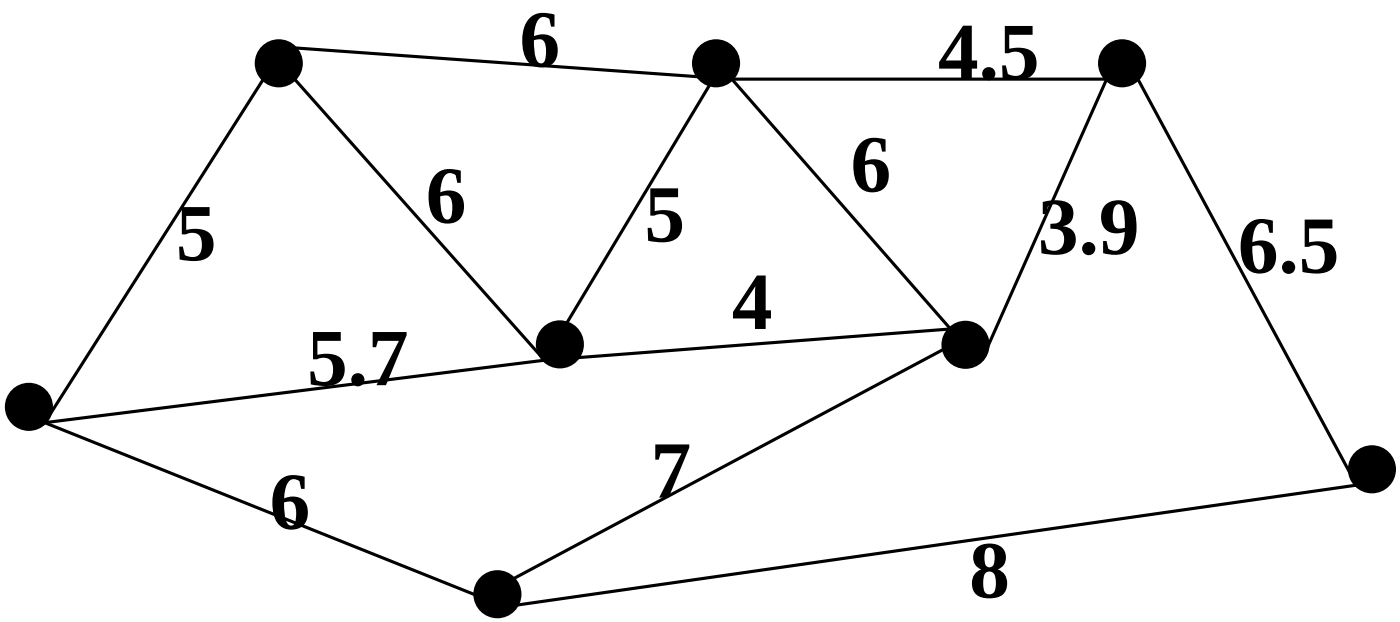


Berechnungsproblem: Berechne eine korrekte 3-Färbung.

Das Traveling Salesperson Problem (TSP)

Angabe: Ein Straßennetz G mit Distanzen, Zahl M.

Frage: Gibt es eine "Tour" mit Gesamtlänge $\leq M$?

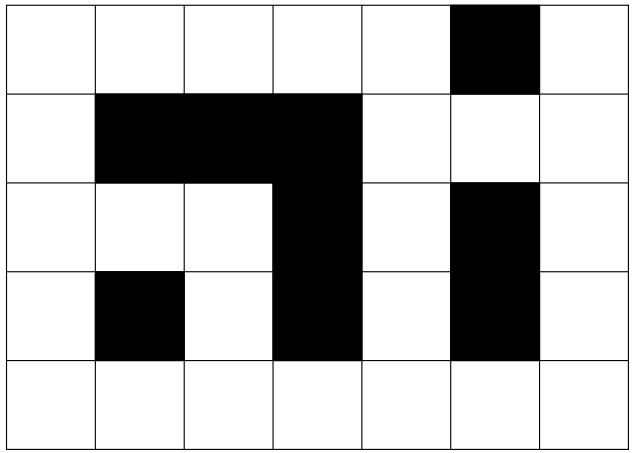


Berechnungsproblem: Berechne optimale Tour.

Kombinatorisches Kreuzworträtsel

Angabe: Raster für Kreuzworträtsel, Wörterbuch.

Frage: Kann ich das Raster füllen?

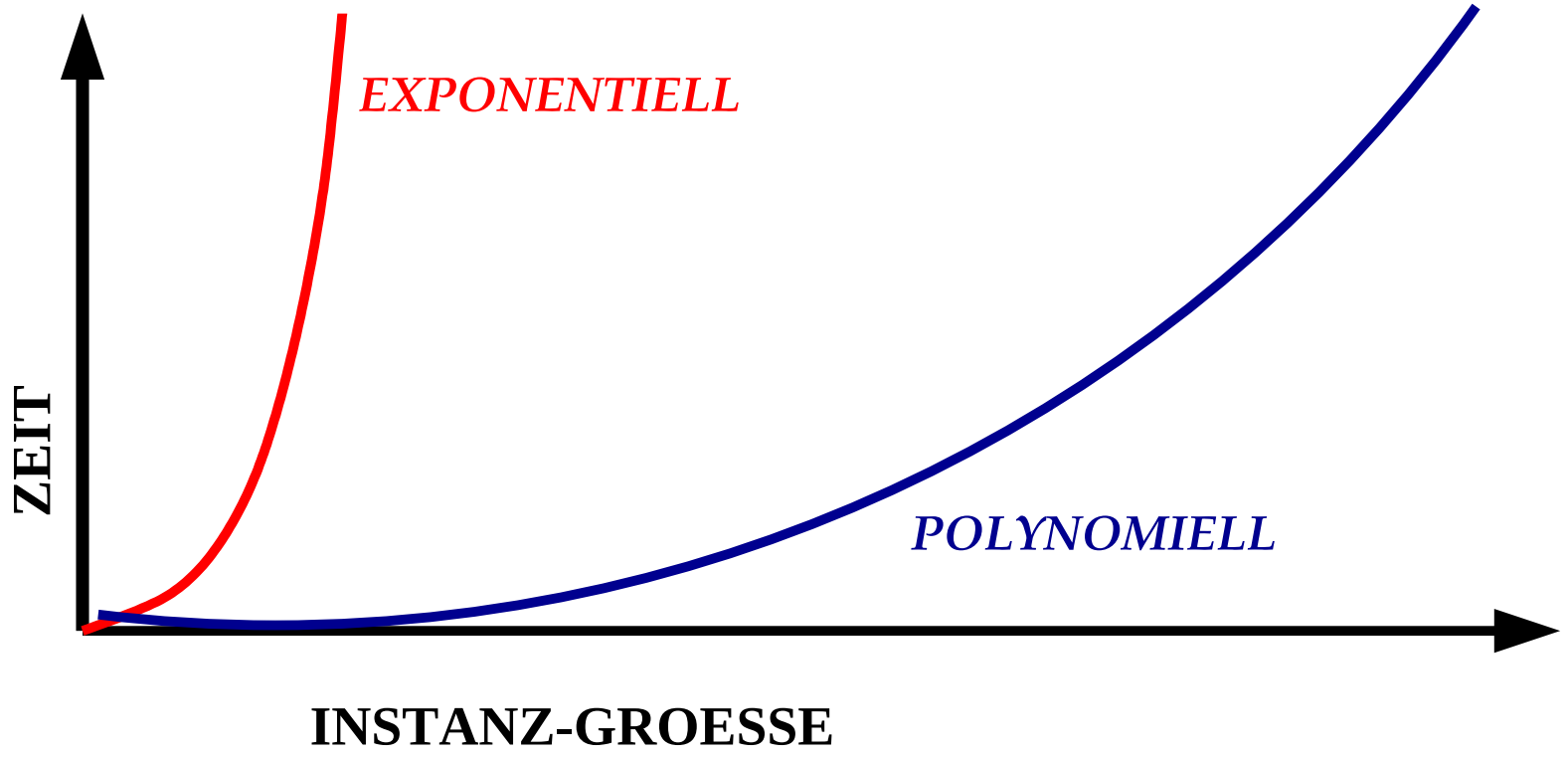


WOERTERBUCH

AUTOMAT
BERGSTEIGER
BODENBELAG
CHEMIE
DAMPFWALZE
DARLEHEN
EIGENART
….. ….. …..

Zeitkomplexität von Algorithmen

Anzahl der Rechenschritte, die ein Algorithmus zur Lösung eines Problems braucht (worst case).

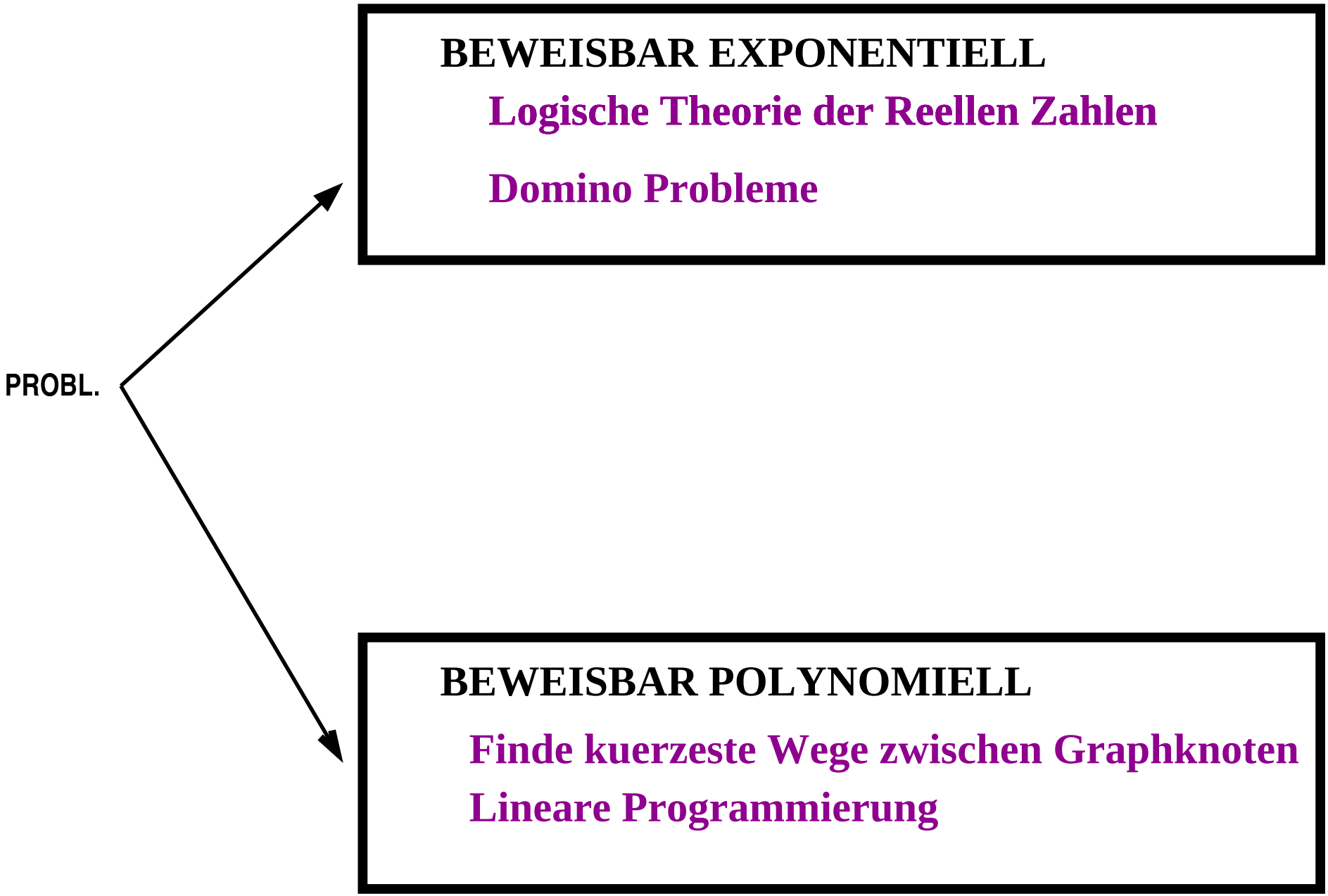


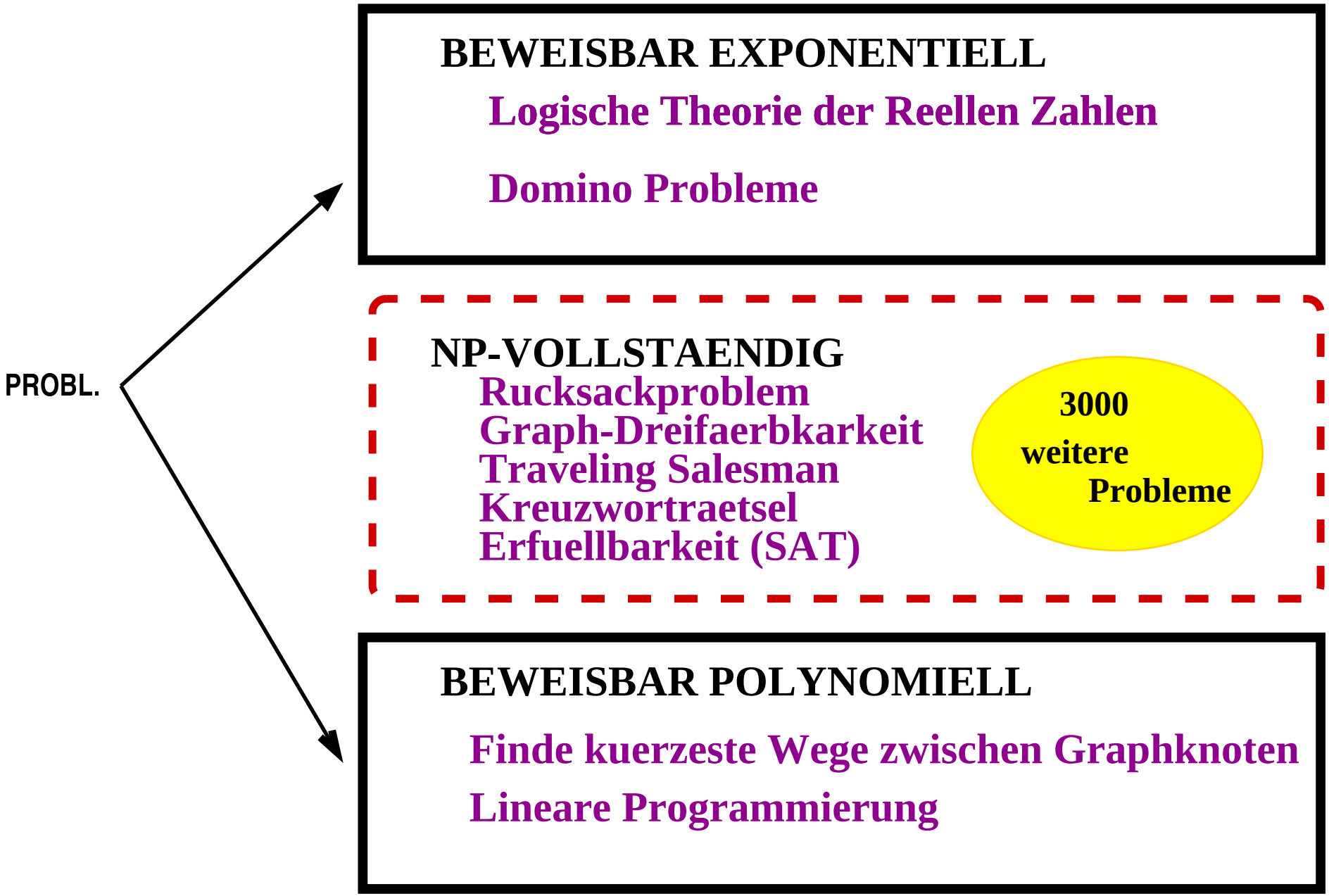
Inhärente Problemkomplexität

Probleme *entscheidbar* oder *unentscheidbar*.

Wir betrachten hier nur entscheidbare Probleme.

Ein Problem ist so komplex wie der bestmögliche Algorithmus zu seiner Lösung.

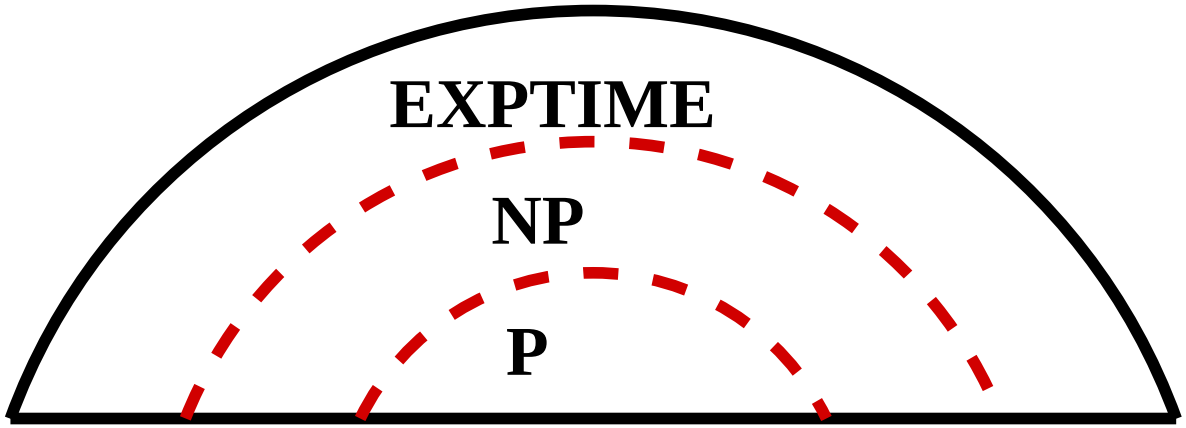

PROBL.
BEWEISBAR EXPONENTIELL
Logische Theorie der Reellen Zahlen
Domino Probleme
BEWEISBAR POLYNOMIELL
Finde kuerzeste Wege zwischen Graphknoten
Lineare Programmierung


PROBL.
BEWEISBAR EXPONENTIELL
Logische Theorie der Reellen Zahlen
Domino Probleme
NP-VOLLSTAENDIG
Rucksackproblem
Graph-Dreifaerbkarkeit
Traveling Salesman
Kreuzwortraetsel
Erfuellbarkeit (SAT)
3000
weitere
Probleme
BEWEISBAR POLYNOMIELL
Finde kuerzeste Wege zwischen Graphknoten
Lineare Programmierung

NP

NP: Nichtdeterministisch polynomiell

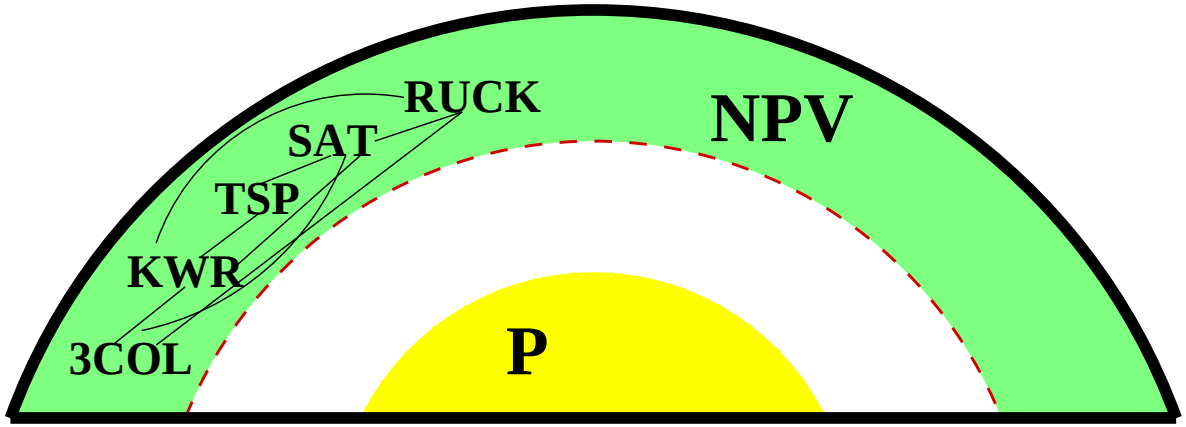
Paradigma: Guess and Check



Kein Problem in NP konnte bisher als nichtpolynomiell bewiesen werden.

Für viele Probleme in NP kennt man aber keinen polynomiellen Algorithmus.

NP-vollständige Probleme



Feinstruktur der Klasse NP

NPV: Die schwersten Probleme innerhalb von NP.

Alle polynomiell ineinander überführbar

Eines polynomiell $\Rightarrow$ alle polynomiell, d.h. NP=P.

Lösungsansätze

NP-vollständige Probleme treten in der Praxis häufig auf.

Sie müssen mit akzeptablen Methoden gelöst werden.

Drei Ansätze:

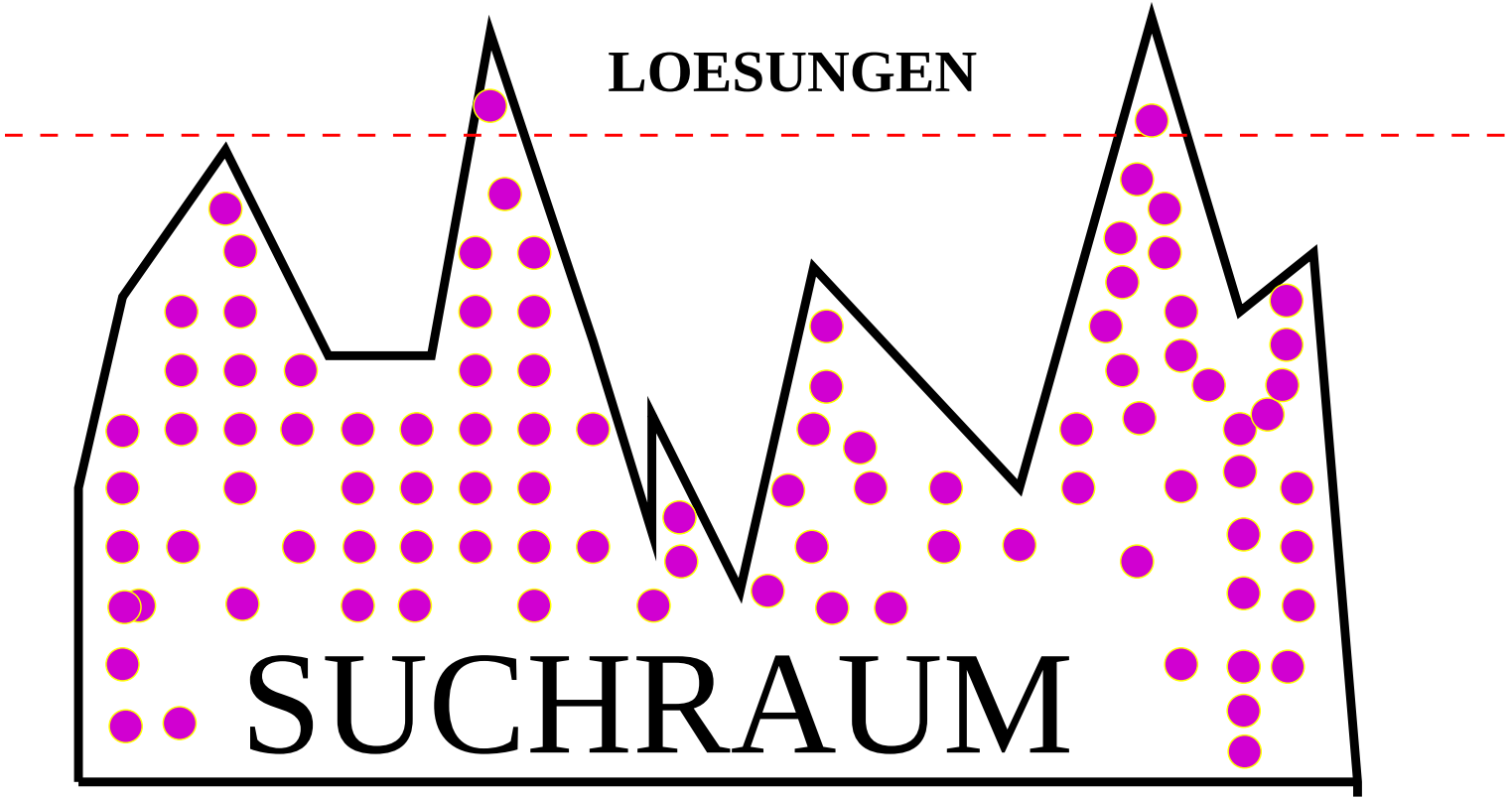
- Randomisierte lokale Suche
- Approximation
- Identifikation leicht lösbarer Subklassen

Randomisierte lokale Suche

Vorgegeben: Bewertungsfunktion für Lösungskandidaten; Zeitlimit.

Verfahren:

1. Erzeuge zufälligen Lösungskandidaten.
2. Führe solange wie möglich lokale Verbesserungen durch.
3. Wenn Lösung gefunden: Ausgeben und Programm beenden.
4. Wenn Zeitlimit erreicht, Abbruch: “Keine Lösung gefunden”
5. Gehe zu Schritt 1.



Beispiel für Bewertungsfunktion (bei Dreifärbbarkeit): Anzahl der korrekt gefärbten Kanten.

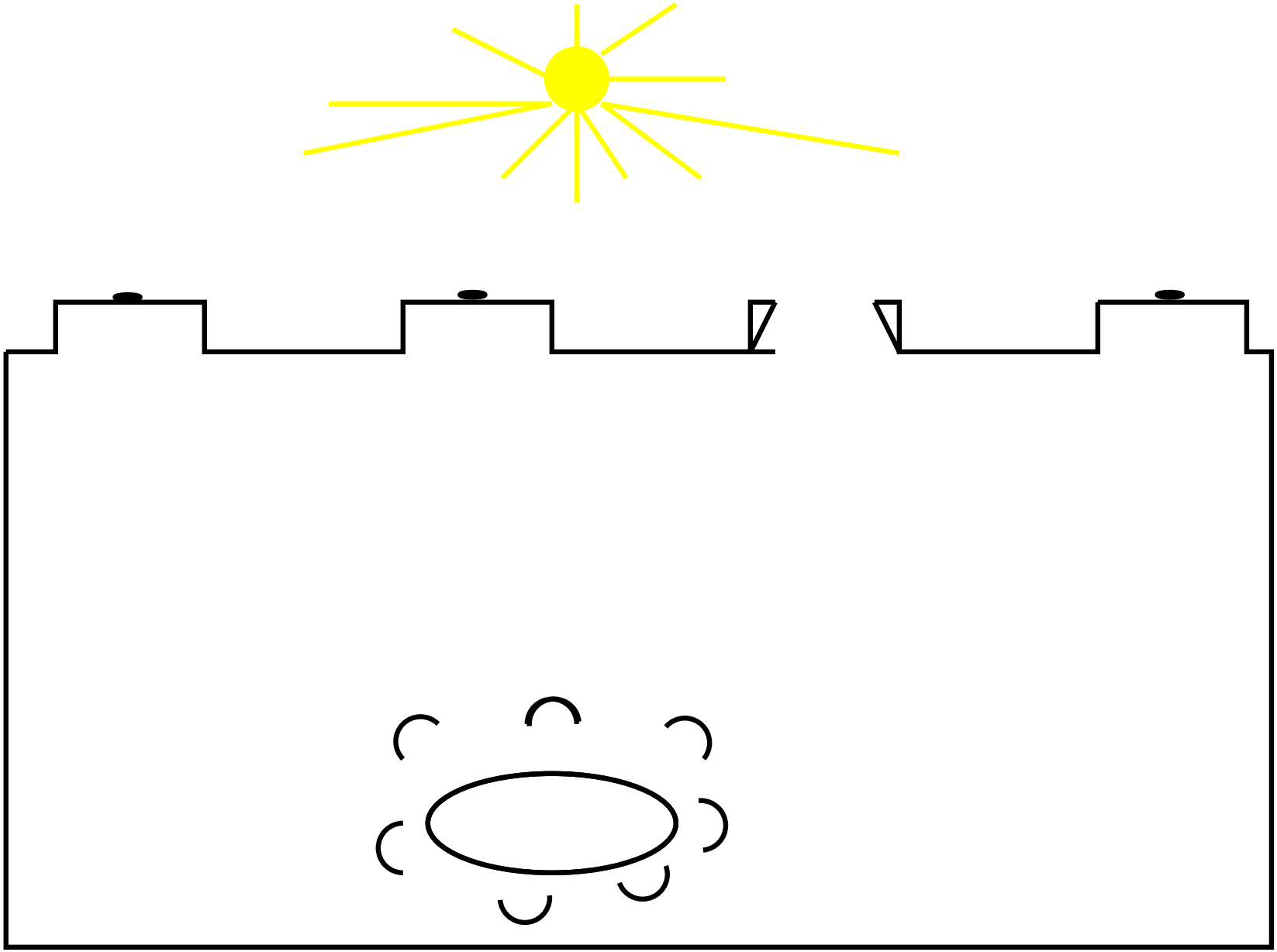
Verfeinerungen: Gradientensuche, Tabusuche, Simuliertes Ausglühen, genetische Algorithmen, usw.

Vorteile der randomisierten lokalen Suche:

- Funktioniert erstaunlich gut für viele Probleme von Praktischer Relevanz.
- Führt dann mit hoher Wahrscheinlichkeit zum Erfolg.
- Liefert vollständige Lösungen.
- Ist ein natürliches Verfahren.

Nachteile:

- Funktioniert schlecht, falls keine Lösung existiert.
- Liefert keinen “Unlösbarkeitsbeweis”
- Funktioniert schlecht bei Instanzen mit wenigen Lösungen.
- Funktioniert nur mit Bewertungsfunktion. (Z.B. nicht geeignet zum Knacken von codes.)



Approximation von Berechnungsproblemen

Ziel: Finde annähernd optimale Lösung bzgl. Bewertungsfunktion w.

Ein Maximierungsproblem ist ε-approximierbar, wenn es es einen Algorithmus T mit polynomieller Laufzeit gibt, sodaß:

Für alle Instanzen x

$$\frac{maxw(x) - w(T(x))}{maxw(x)} \leq \varepsilon$$

Approximationsgrad $\gamma(A)$ eines Maximierungsproblems A: Größte untere Schranke für ε-Approximierbarkeit.

$$\gamma(A) = \inf\{\varepsilon \mid A \text{ ist } \varepsilon - \text{approximierbar}\}.$$

Beispiele

- $\gamma(\text{RUCKSACK}) = 0$; beliebig approximierbar.
- $\gamma(\text{TSP}) = 1$ außer P=NP; überhaupt nicht approximierbar.
- $\gamma(\text{TSP}/\text{EUCLID.}) \leq \frac{1}{3}$.

$\gamma(A) = 0$ bedeutet nicht, daß A polynomiell lösbar!

Vorteil der Approximationsmethode:

- Approximation in der Praxis oft ausreichend (brauche keine exakte Lösung).
- Auch auf manche beweisbar exponentielle Probleme anwendbar.

Nachteile:

- Wenige praktische Probleme approximierbar.
- Nicht auf Entscheidungsprobleme anwendbar.

Approximationstheorie: Eleganter und reichhaltiger Zweig der theoretischen Informatik. (Doz. Wöginger, TU Graz).

Identifikation von Polynomiellen Subklassen

Hohe Komplexität oft nur in “worst cases” gegeben.

Vertrackte Struktur der worst case Probleminstanzen.

Für Eingaben einfacherer Struktur würde polynomieller Algorithmus existieren.

In der Praxis sind viele Eingabeinstanzen einfach.

Daher:

- Definiere geeignete polynomiell lösbare Subklassen v. Instanzen.
- Zeige, daß Zugehörigkeitstest polynomiell ist.
- Entwickle gute polynomielle Algorithmen für diese Klassen.

Beschränkte Baumweite (bounded treewidth)

Bei graphbasierten Problemen ist hohe Komplexität meist durch Zyklizität bedingt.

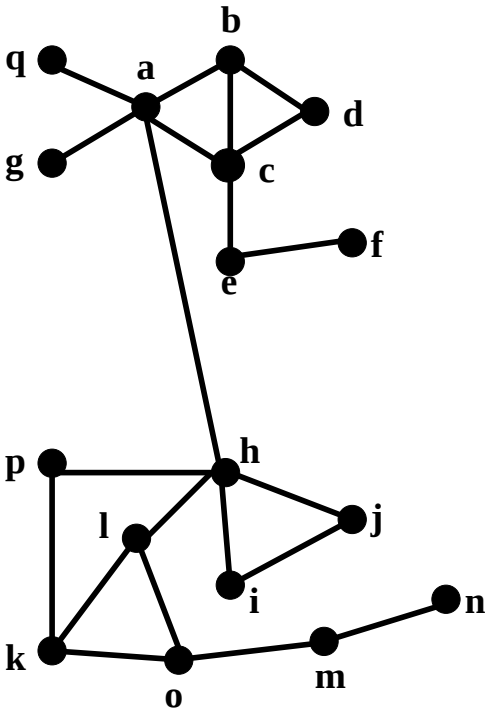
Für *azyklische* Graphen sind Probleme oft trivial lösbar. ($\rightarrow$3COL)

Viele Graphenprobleme sind auch für Instanzen *niederer Zyklizität* lösbar.

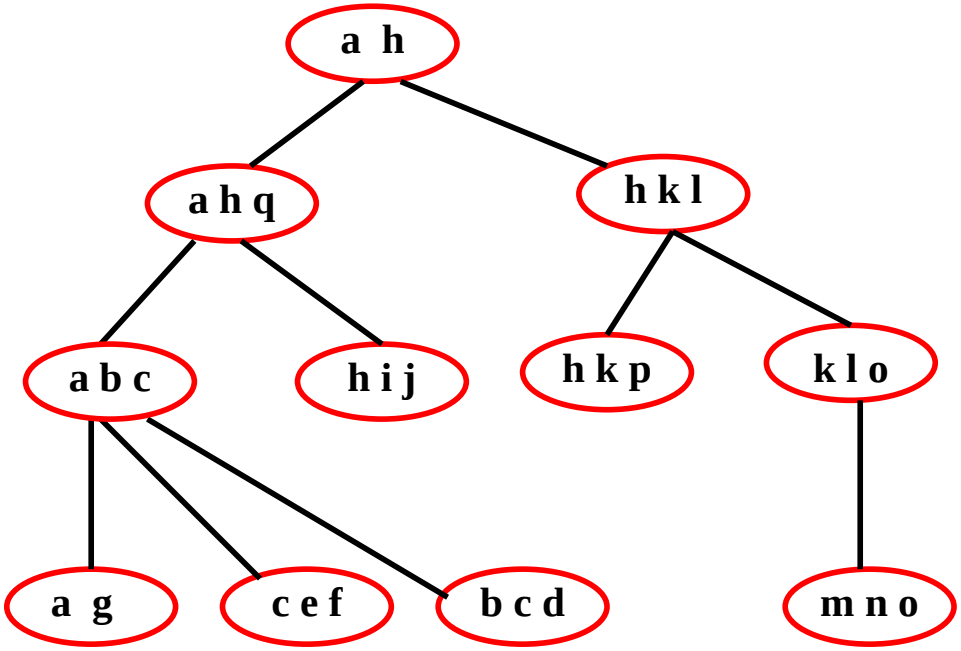
Maß für den Zyklizitätsgrad eines Graphen?

Baumweite (Robertson und Seymour)

Basiert auf dem Konzept einer Baumdekomposition.



Graph G



Tree decomposition of width 2 of G

Baumweite eines Graphen: Kleinstmögliche Weite von Baumzerlegung.

$bw(G)$

Satz. (Bodlaender): Für fixe Konstante k kann man in linearer Zeit feststellen, ob $bw(G) \leq k$.

Viele NP-vollständige Graphprobleme sind für Graphen mit konstant beschränkter Baumweite in linearer Zeit lösbar.

Bsp. Dreifärbbarkeit (3COL).

Verbindung zur Logik

Satz. (Fagin): Jedes NP-Problem über Graphen läßt sich durch eine existentielle Formel der Prädikatenlogik zweiter Stufe (SO) darstellen.

NP=ESO

Monadisches SO (MSO): Teilklasse von SO, nur Mengenvariablen, aber keine mehrstlligen Relationenvariablen zugelassen.

Dreifärbbarkeit $\in$ MSO.

Satz. (Courcelle): Alle Graphprobleme, die sich in MSO darstellen lassen, sind polynomiell für Graphen mit konstant beschränkter Baumweite.

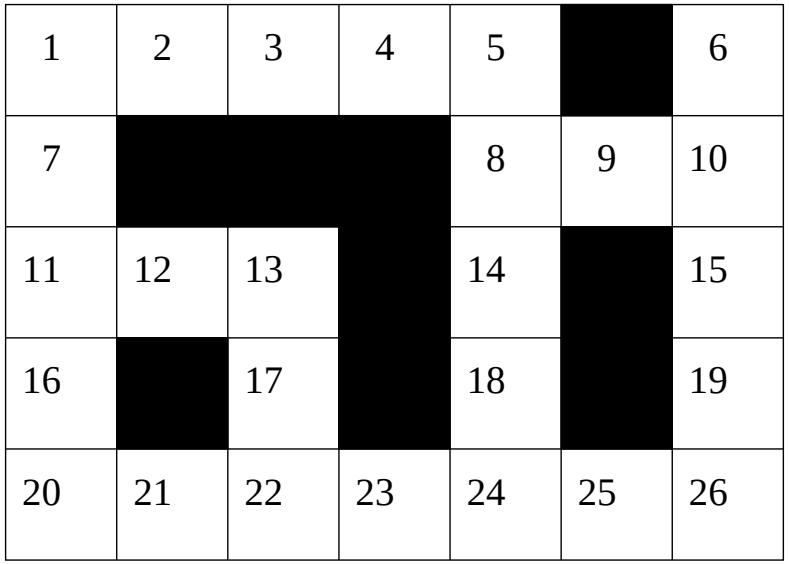
Dreifärbbarkeit als MSO-Formel

Der Graph wird als Struktur $\langle U, E \rangle$ angesehen.

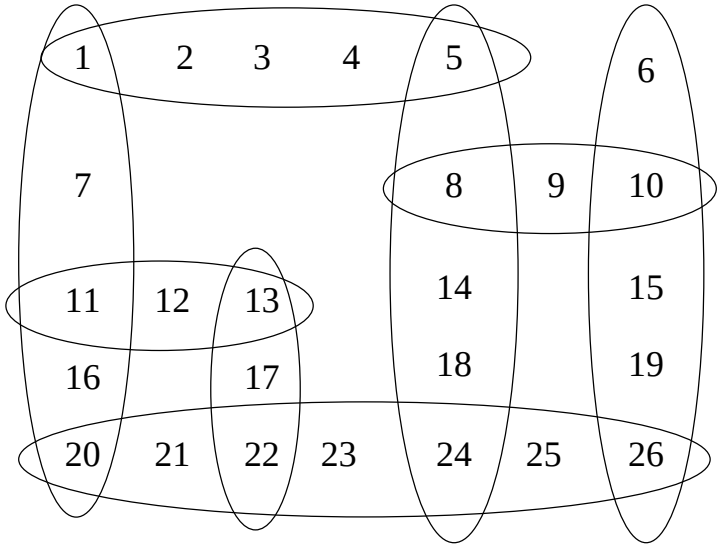
U: Universum, Menge der Knoten; E: Kanten, 2stellige Relation.

$$
\begin{aligned}
(\exists R, G, B) \quad [\quad & (\forall x\,(R(x) \vee G(x) \vee B(x))) \\
\wedge \quad & (\forall x(R(x) \Rightarrow (\neg G(x) \wedge \neg B(x)))) \\
\wedge \quad & \ldots \\
\wedge \quad & \ldots \\
\wedge \quad & (\forall x,y(E(x,y) \Rightarrow (R(x) \Rightarrow (G(y) \vee B(y))))) \\
\wedge \quad & (\forall x,y(E(x,y) \Rightarrow (G(x) \Rightarrow (R(y) \vee B(y))))) \\
\wedge \quad & (\forall x,y(E(x,y) \Rightarrow (B(x) \Rightarrow (R(y) \vee G(y)))))\,]
\end{aligned}
$$

Viele Probleme lassen sich besser durch *Hypergraphen* als durch Graphen beschreiben.



CSP instance I



Hypergraph $\mathcal{H}_I$

Dekomposition von Hypergraphen

Dzt. intensives Forschungsgebiet. Vorschläge:

- *Hypergraph-Dekomposition/-Weite* Robertson und Seymour.
- *Query-Dekomposition/-Weite* Chekuri und Rajaraman.

Satz. (Chekuri und Rajaraman): Hypergraphbasierte CSP- Probleme sind polynomiell, wenn Hypergraph-Weite konstant beschränkt.

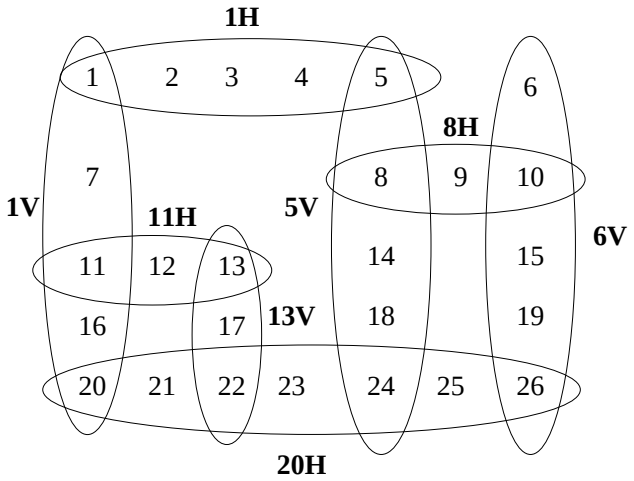
Frage: Kann man Hypergraphen von beschr. Weite leicht erkennen?

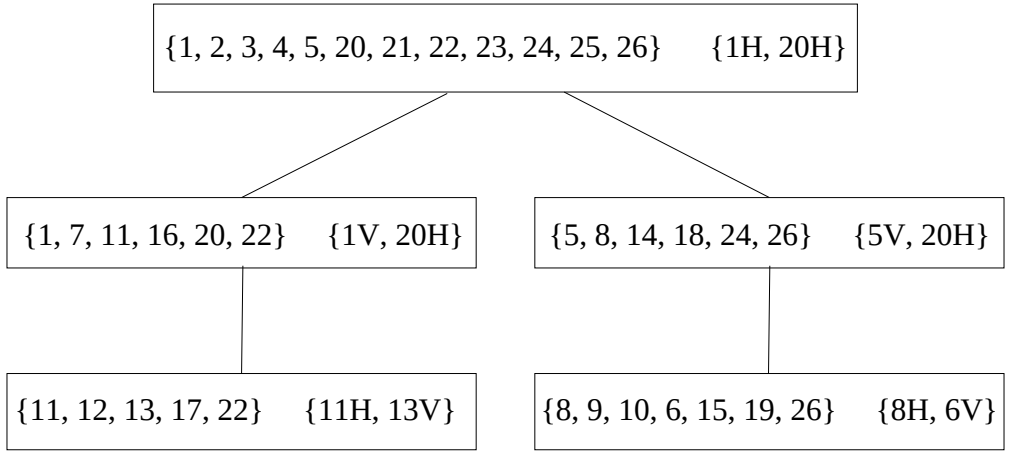
Antwort: nein!

Satz. (G., Leone, Scarcello): Für $k > 3$ ist dieses Erkennungsproblem NP-vollständig.

Hypertree Decompositions

Neuer Ansatz zur Hypergraphendekomposition.





Hypergraph $\mathcal{H}_I$

A hypertree decomposition of width 2

Fakten zur Hypertree-Weite

Satz. Hypergraphbasierte CSP-Probleme sind polynomiell, wenn Hypertree-Weite konstant beschränkt.

Satz. Für jede fixe Konstante k kann in polynomieller Zeit festgestellt werden, ob ein Hypergraph durch k beschränkte Hypertree-Weite hat.

Satz. $\forall G$: Hypertree-Weite(G) $\leq$ Query-Weite(G).

Daher ist unser Ansatz allgemeiner und umschließt mehr Instanzen.

Vorteile der Verwendung solcher polynomieller Subklassen

- In vielen Anwendungsbereichen haben fast alle “natürlichen” Eingaben Hypertree-Weite ≤ 5.
- Auch negative Instanzen werden erkannt.
- Hoher Parallelisierungsgrad der Subklassen (LOGCFL).

Nachteile:

- Sagt nichts über Instanzen von großer Weite.
- Bei hypergraphbasierten Methoden steigt Polynomgrad mit Baumweite.

Schluß

- Was ist ein Entscheidungsproblem / Berechnungsproblem ?
- Komplexität von Algorithmen und Problemen
- NP-vollständige Probleme
- Drei Lösungsansätze:
  - Randomisierte lokale Suche
  - Approximation
  - Identifikation leicht lösbarer Subklassen

In allen Teilbereichen offene Probleme. Aktive Forschung.

Quantencomputer: Nicht zur polynomiellen Lösung von NPV-Probleme

geeignet; aber für Subklasse RP von NP.