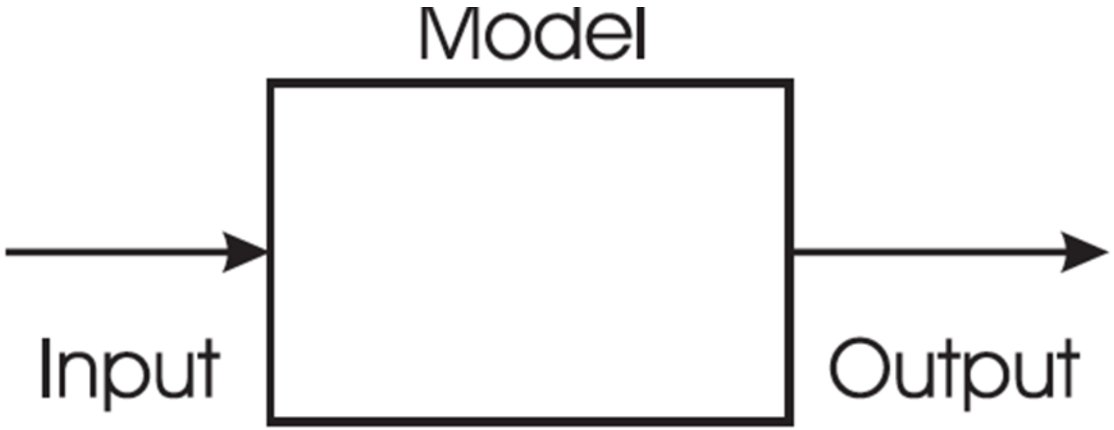
Klassifikation von Problemen

Jens Kosiol

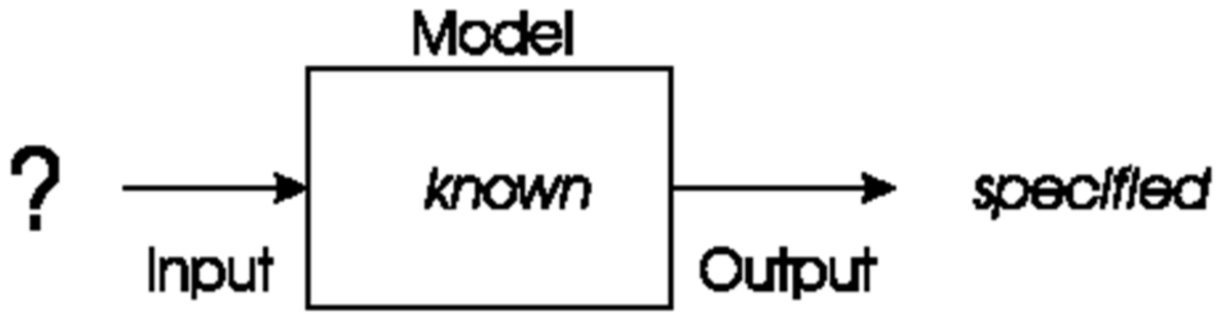
(Foliensatz basiert weitgehend auf dem entsprechenden Foliensatz von Eiben und Smith)

Black-Box Modell für Berechnungen



Verschiedene **Arten von Problemen** entstehen, je nachdem welche der drei Größen unbekannt ist

Optimierungsproblem



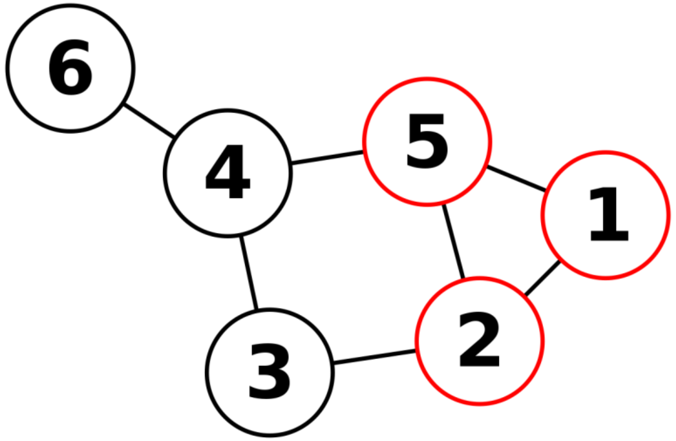
- Model ist bekannt
- Gewünschtes Ergebnis ist bekannt (zumindest eine Beschreibung davon)
- Aufgabe: Finde eine Eingabe, für die das Modell das gewünschte Ergebnis ausgibt.

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Cliquenproblem (Optimierungsversion)

Gegeben: Ein Graph

- Input: Untergraphen dieses Graphen
- Modell: Ein Algorithmus, der die Größe des Untergraphen bestimmt (Knotenzahl) und entscheidet, ob es sich bei dem Untergraphen um eine *Clique* (vollständiger Teilgraph) handelt
- Output: Ja oder nein und die Größe des Untergraphen



- Gesucht: Input, der größtmögliche Ja-Instanz ist

Traveling Salesman Problem (Optimierungsversion)

Gegeben: Ein gewichteter vollständiger Graph

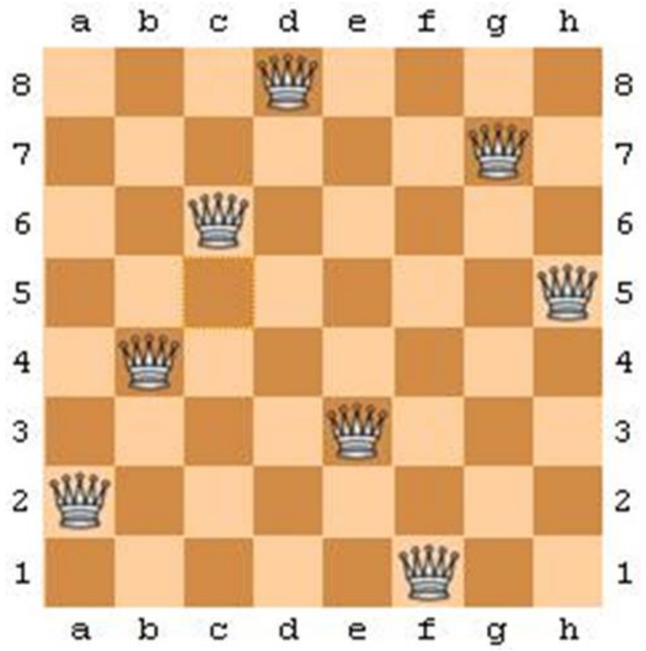
- Input: Tour durch den Graphen (Permutation der Knoten des Graphen)
- Modell: Ein Algorithmus, der die Länge der Tour berechnet (Summe der Gewichte der Wege, die zu der Tour gehören)
- Output: Länge der Tour

- Gesucht: Input, der kürzeste Tour liefert

Damenproblem (Optimierungsversion)

Gegeben: Ein $n \times n$ Schachbrett mit n Damen

- Input: Eine Positionierung der n Damen auf dem Schachbrett
- Modell: Ein Algorithmus, der zählt, wie viele Damen sich gegenseitig schlagen
- Output: Anzahl der Damen, die sich gegenseitig schlagen



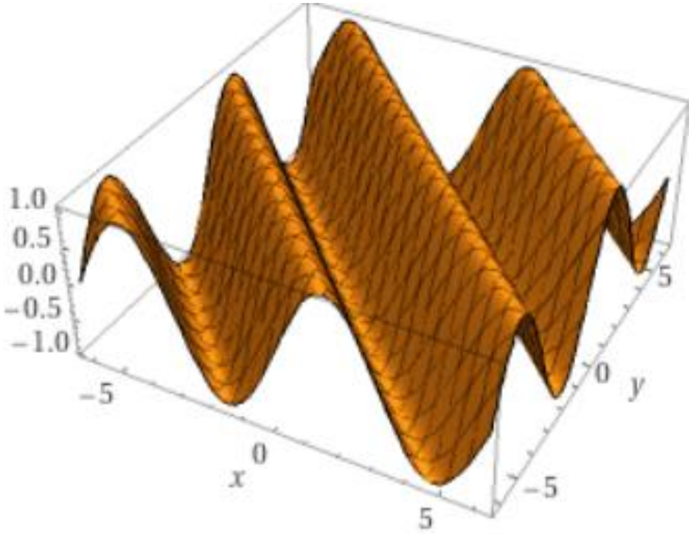
- Gesucht: Eine Positionierung, die zum Ergebnis 0 führt

Optimierung einer Funktion

Gegeben: Eine Funktion $f\colon\ \mathbb{R}^n \to \mathbb{R}$

- Input: Ein n-dimensionaler Vektor $x = (x_1, \ldots, x_n)$
- Modell: Auswertung der Funktion f
- Output: $f(x)$

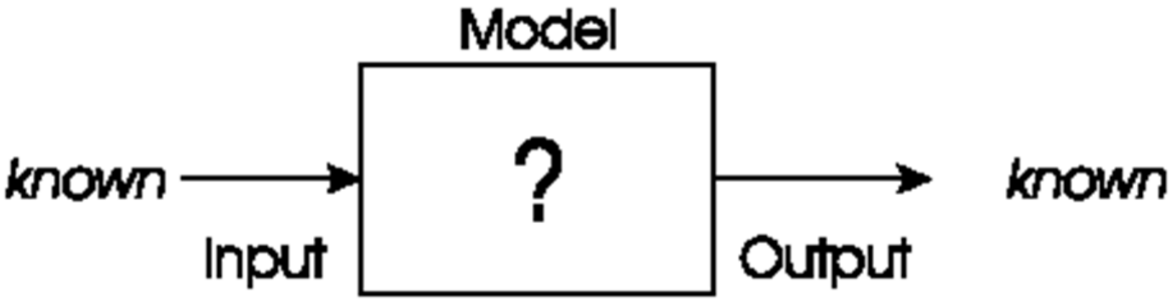
- Gesucht: Eingabe x (aus dem Definitionsbereich D), sodass $f(x)$ minimal/maximal ist



Kern Optimierungsprobleme

- Eine Zielfunktion (auch Bewertungsfunktion, Fitnessfunktion, objective function)
  - Größe einer Clique
  - Kosten einer Tour
  - Anzahl der Damen, die sich schlagen
  - Eine mathematische Funktion
- Ein Suchraum von möglichen Eingabewerten
  - Knotenmenge
  - Rundtour
  - Positionierung von Damen
  - Vektor
- Gesucht: Eingabe, die die Zielfunktion minimiert/maximiert

Modellierungsproblem

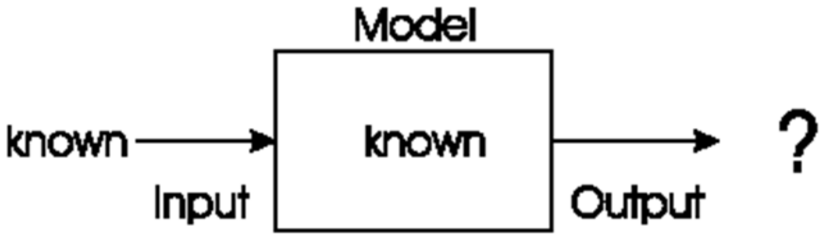


- Ein- und Ausgabepaare sind bekannt
- Aufgabe: Finde ein Modell, das für die bekannten Eingaben die korrekten Ausgaben liefert.

Beispiele für Modellierungsprobleme

- Suche nach Naturgesetzen
- Interpolation (Mathematik)
- Natur- und wirtschaftswissenschaftliche Modellbildung
  - Klimamodelle
  - Modelle für Versicherungen und Banken
  - …
- Entwicklung eines neuronalen Netzes (oder anderen Classifiers)

Simulation/Berechnung



- Eingabe und Modell sind bekannt
- Aufgabe: Berechne Ausgabe

Beispiele Simulation/Berechnung

- Wettervorhersage/Klimavorhersage
- Simulation eines Motors (oder irgendeines Bauteils) mit bestimmten Einstellungen
- Bestimmen der Versicherungssumme, Kreditwürdigkeit, …
- Auswerten einer Funktion
- Klassifikation eines Bildes durch ein neuronales Netz
- …

Besonders interessant, um „Was wenn …?“-Fragen zu beantworten

Vergleich Problemarten

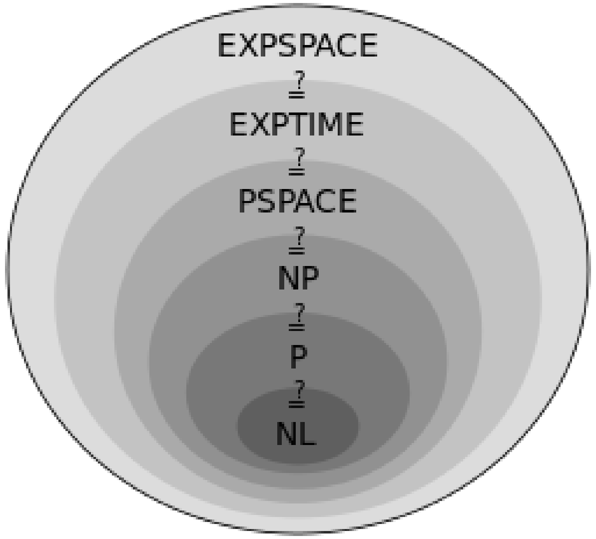
- Simulation/Berechnung erfordert das Schaffen von Simulationsumgebung, Computeralgebrasystemen, …
- Optimierungsprobleme führen zum Konzept des Suchraums: Der (potentiell unendlich große) Raum möglicher Lösungen
- Modellierungsprobleme können ganz unterschiedliche Techniken erfordern (je nach Art des Modells)
- Modellierungsproblem als Optimierungsproblem: Lege Art des gesuchten Modells fest (z. B. ein neuronales Netz) und durchsuche den Raum der so möglichen Modelle nach einem mit minimalem Fehler

Zur Lösung von Optimierungsproblemen brauchen wir Verfahren, um große Suchräume zu durchsuchen!

Schwierigkeit von Problemen

Keine heuristischen Verfahren für einfache Probleme nutzen!

Aber das Nutzen von Approximation und Metaheuristiken ist ein vielversprechender Ansatz für harte Probleme.



Struktur von Optimierungsproblemen/Suchräumen

- Suchen von Optima differenzierbarer Funktion
- Konvexer Suchraum
- Lineare Optimierung
- Unimodalität
- …

Wenn das Problem eine bekannte Struktur hat, sollte man diese ausnutzen!

Optimierung vs. Constraint Satisfaction

- Optimierungsproblem: Suche nach Eingabewert, der Zielfunktion maximiert/minimiert; Intuition: Zielfunktion misst Güte der Lösung
- Constraint-Satisfaction-Problem (CSP)
  - Gegeben ist eine Menge von Constraints (Bedingungen): boolesche Ausdrücke über einer Menge von Variablen
  - Gesucht wird eine Variablenbelegung, sodass jede Bedingung zu „wahr“ auswertet
  - Variablenbelegung bildet entweder Lösung oder nicht

Beispiele CSPs

- Finde eine Konfiguration von 8 Damen auf einem Schachfeld, sodass keine zwei sich schlagen
- SAT-Problem
- Färbung eines Graphen
- …

Kombination Optimierung und CSP

	Objective function	
Constraints	**Yes**	**No**
Yes	Constrained optimisation problem	Constraint satisfaction problem
No	Free optimisation problem	No problem

- TSP mit Nebenbedingung, dass Stadt X vor Stadt Y besucht wird
- Optimierung einer mathematischen Funktion auf eingeschränktem Definitionsbereich
- Rucksackproblem

Formalisierungen Damenproblem

Als Constraint Satisfaction Problem:

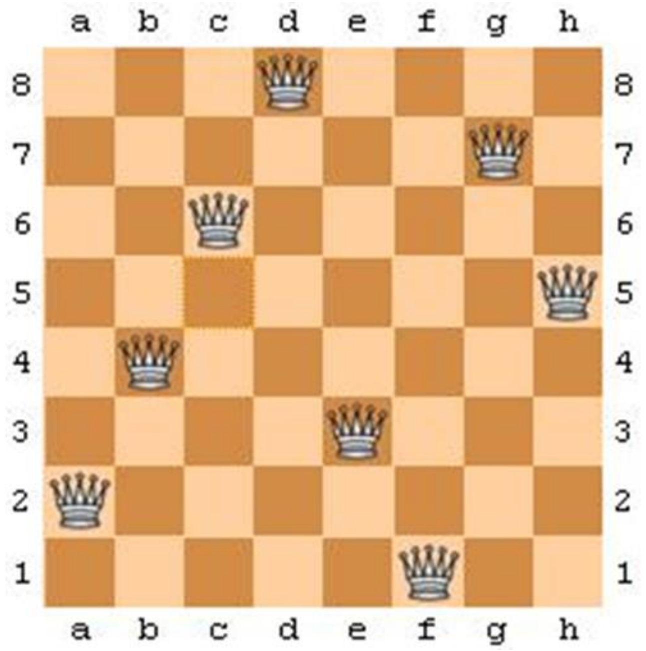
- Für jedes Feld des Schachbretts eine Formel, die sagt, dass, falls dieses Feld mit einer Dame besetzt ist, keins der durch die Dame erreichbaren Felder besetzt ist.
- Eine Formel, die ausdrückt, dass n Felder mit Damen besetzt sind

Als freies Optimierungsproblem:

- Maximiere die Funktion f, die zählt, wie viele Damen keine andere schlagen

Als beschränktes Optimierungsproblem:

- Für jedes Feld des Schachbretts eine Formel, die sagt, dass, falls dieses Feld mit einer Dame besetzt ist, keine weitere Damen in der entsprechenden Spalte und Reihe stehen.
- Funktion g, die zählt, wie viele Damen keine andere diagonal schlagen



Evolutionäre Algorithmen als Lösungsstrategie

- Metaheuristische Methode zum Erkunden eines Suchraums:
  - Heuristisch: Liefert nicht garantiert eine optimale Lösung, sondern approximiert eine solche
  - „Meta“: Prinzipiell unabhängig von der Zielfunktion, die optimiert werden soll
- Können mit freien und mit Optimierungsproblemen mit Nebenbedingungen umgehen
- Sollten auf schwere Probleme angewendet werden, bei denen nicht (oder nur schwer) eine vorhandene Struktur ausgenutzt werden kann