Computational Social Choice

Felix Brandt
Institut für Informatik
LMU München

PREFERENCE AGGREGATION IN MULTIAGENT SYSTEMS

Organisatorisches

- Vorlesung
  - Mittwoch, 10.15-11.45 Uhr, Raum 0.41
- Keine Übung
  - Viele Beispiele in der Vorlesung
- Bei Bedarf Vorlesungsschein durch erfolgreiche mündliche Prüfung
- Nächste Woche (24.10.) keine Vorlesung
  - Stattdessen Vortrag TUM morgen (18.10.), 11.00 Uhr, Raum 03.09.014
    Ariel Procaccia (The Hebrew University of Jerusalem):
    "Algorithms for the Coalitional Manipulation Problem"

Literatur

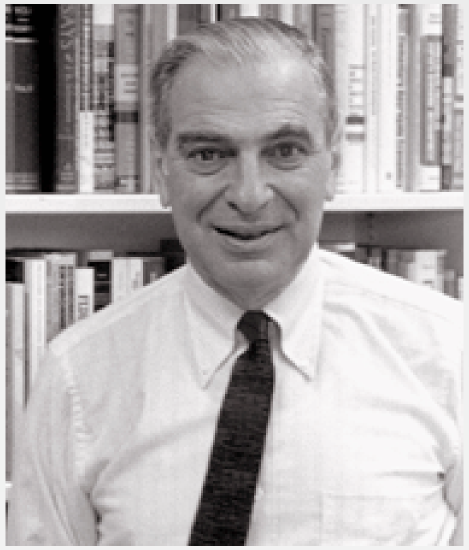
- Einführend
  - Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A Short Introduction to Computational Social Choice. LNCS 4362, Springer-Verlag, 2007.
  - U. Endriss and J. Lang. Proceedings of the 1st International Workshop on Computational Social Choice, ILLC, 2006.
  - P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A Richer Understanding of the Complexity of Election Systems, 2007.
- Weiterführende Bücher
  - M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
  - J. Laslier. Tournament Solutions and Majority Voting. Springer-Verlag, 1997.
  - H. Moulin. Axioms of Cooperative Decision Making. Cambridge University Press, 1988.

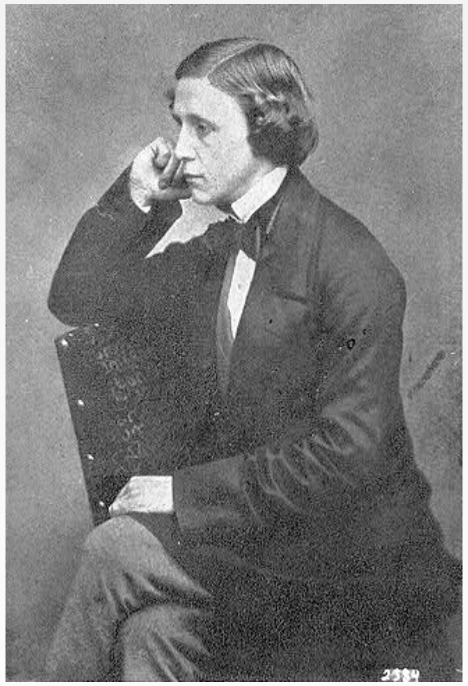
Motivation

- Was ist “social choice theory”?
    - Hauptproblem
        - Vorhanden: Viele unterschiedliche Meinungen
        - Gesucht: Eine gemeinsame Entscheidung
    - Theorie kollektiver Entscheidungen
    - Forschungsgebiet zwischen Mathematik und Ökonomie (ähnlich Spieltheorie)
- Wichtige Bestandteile
    - Autonome Agenten (z.B. Menschen oder Computerprogramme)
    - Alternativen (meistens endlich viele)
    - Präferenzen über Alternativen
    - Bündelungsfunktionen

Wichtige Figuren

- Amartya Sen
  - Nobelpreis 1998
- Kenneth J. Arrow
  - Arrows Unmöglichkeitssatz
  - Nobelpreis 1972
- John George Kemeny
  - 1926-1992
  - Erfinder der Programmiersprache Basic
- Charles Dodgson (Lewis Carroll)
  - 1832-1898
- Marie Jean Antoine Nicolas Caritat (Marquis de Condorcet)
  - 1743-1794





Computational Social Choice

- Anwendungsbeispiele:
  - Wahlverfahren (für Menschen oder Software-Agenten)
  - Internetseitenbewertung (z.B. PageRank)
  - kollaborative Bewertungssysteme (z.B. amazon oder ebay)
- Hauptsächlicher Inhalt dieser Vorlesung
  - Algorithmische Aspekte von Bündelungsfunktionen
- Andere Aspekte (nicht Teil dieser Vorlesung)
  - Berechnungskomplexität strategischen Wahlverhaltens (morgen TUM)
  - Gerechte Aufteilung (fair division)
  - Kommunikationskomplexität und Datenschutz

Effiziente Algorithmen (1)

- Eine der ... hmus ist **Zeit**.
- Wichtig... polynom...
  - Wenn j... Algorithmus?
- Warum ...

Time complexity function	Size n: 10	20	30	40	50	60
n	.00001 second	.00002 second	.00003 second	.00004 second	.00005 second	.00006 second
n^2	.0001 second	.0004 second	.0009 second	.0016 second	.0025 second	.0036 second
n^3	.001 second	.008 second	.027 second	.064 second	.125 second	.216 second
n^5	.1 second	3.2 seconds	24.3 seconds	1.7 minutes	5.2 minutes	13.0 minutes
2^n	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	.059 second	58 minutes	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries

Figure 1.2 Comparison of several polynomial and exponential time complexity functions.

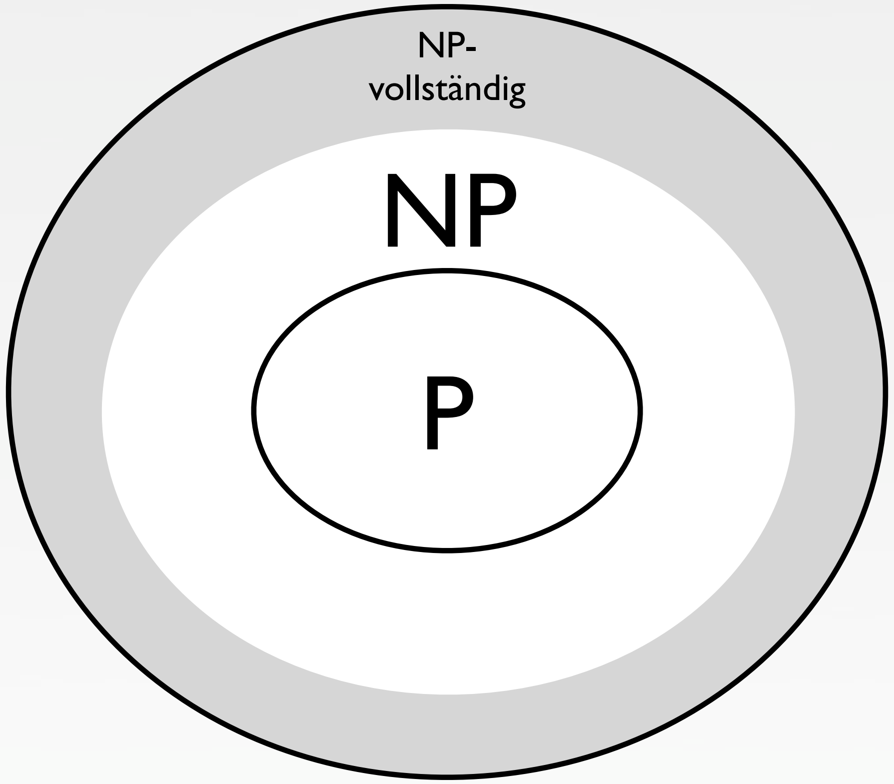
[Garey & Johnson, 1979]

Effiziente Algorithmen (2)

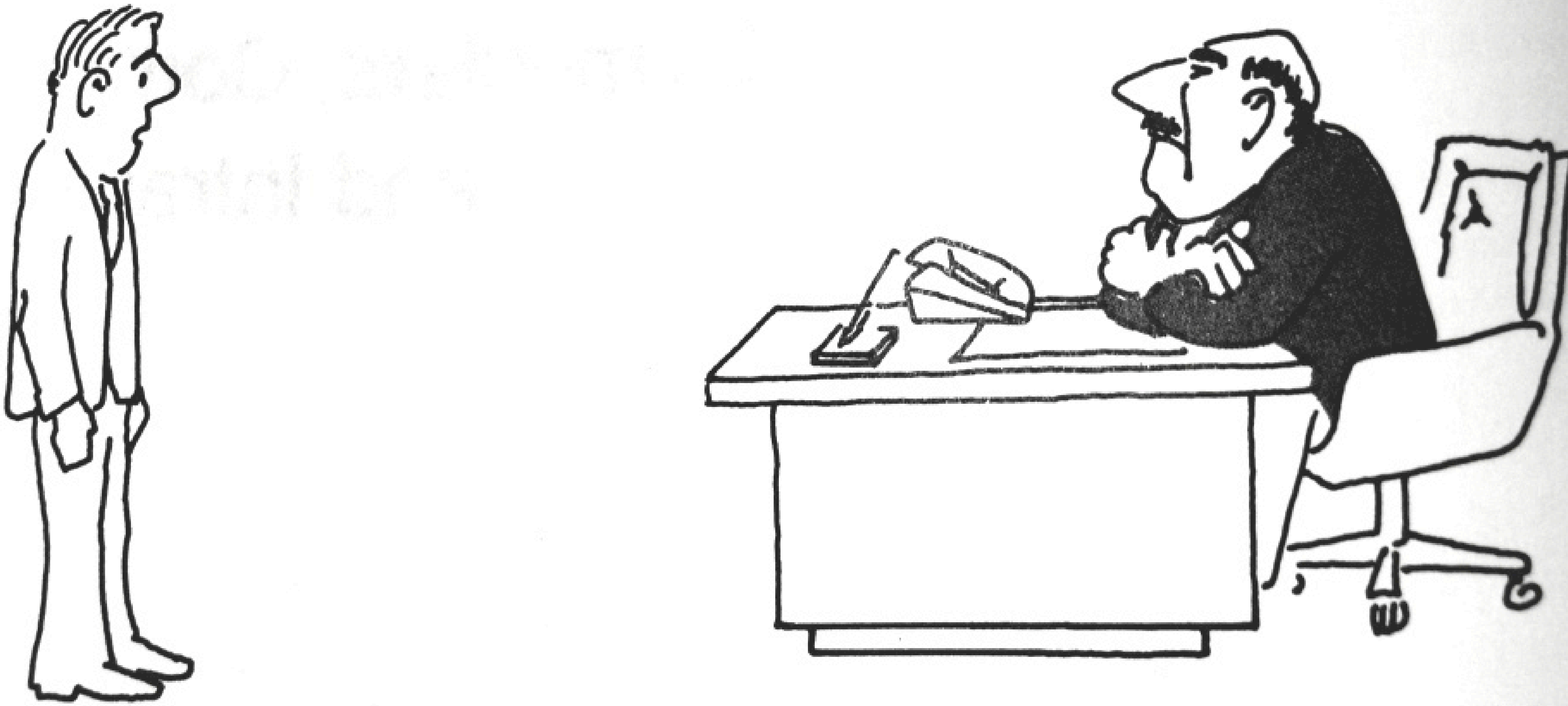
- Wie findet man einen effizienten Algorithmus?
  - Häufig mit Standardtechniken wie
    - greedy,
    - divide and conquer,
    - dynamische Programmierung,
    - lineare Optimierung,
    - usw.
- Wie kann man zeigen, dass kein effizienter Algorithmus existiert?
  - In den meisten Fällen gar nicht.
  - Aber häufig kann man eine ähnlich starke Aussage treffen: NP-Schwierigkeit.

Algorithmen und Komplexität

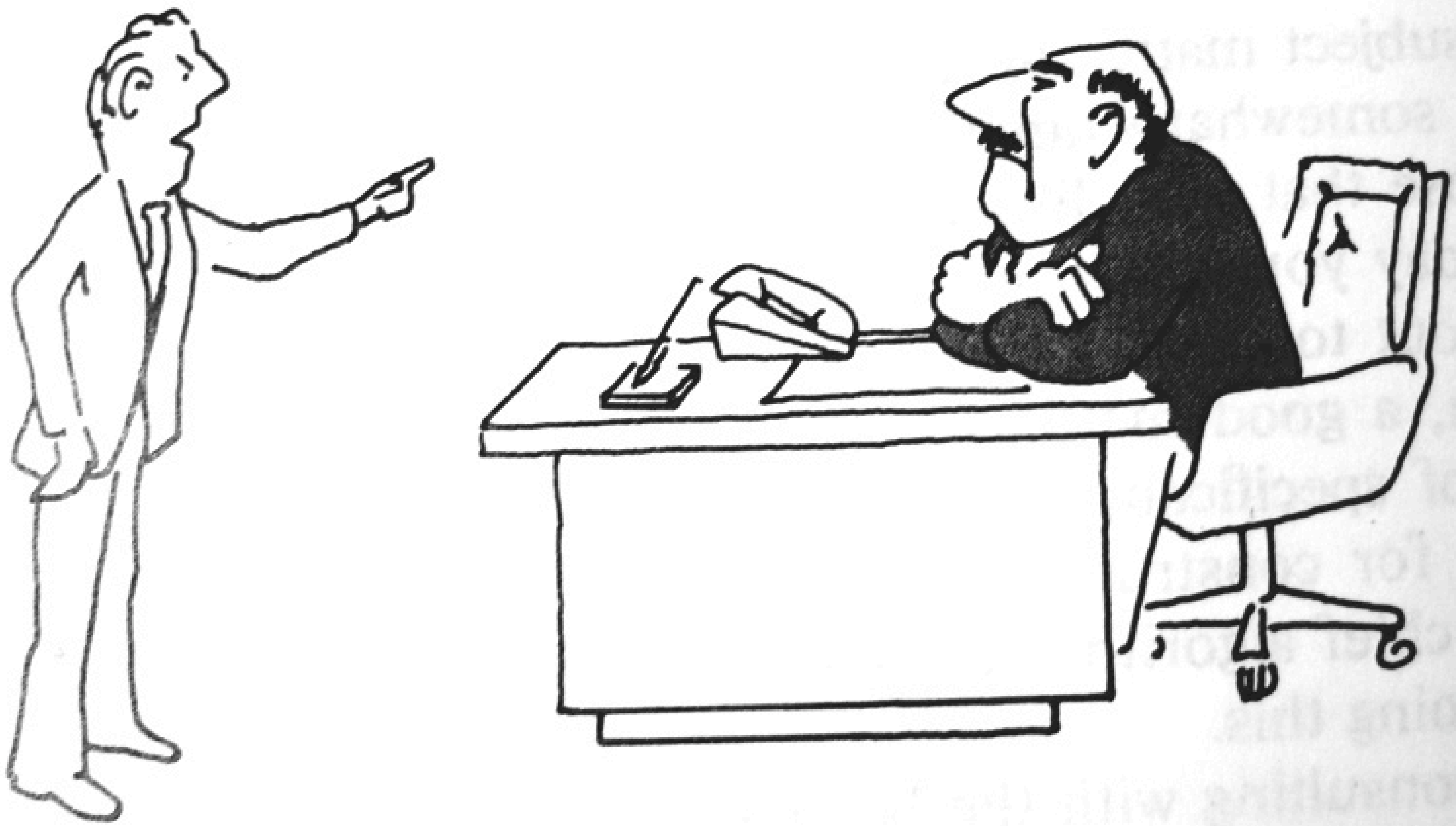
- P (in polynomieller Zeit lösbar auf einer deterministischen Turingmaschine)
  - Es existieren effiziente Algorithmen.
- NP (in polynomieller Zeit lösbar auf einer nicht-deterministischen Turingmaschine)
  - In polynomieller Zeit verifizierbar
  - Für NP-*schwere* Probleme existieren keine effizienten Algorithmen wenn P≠NP.



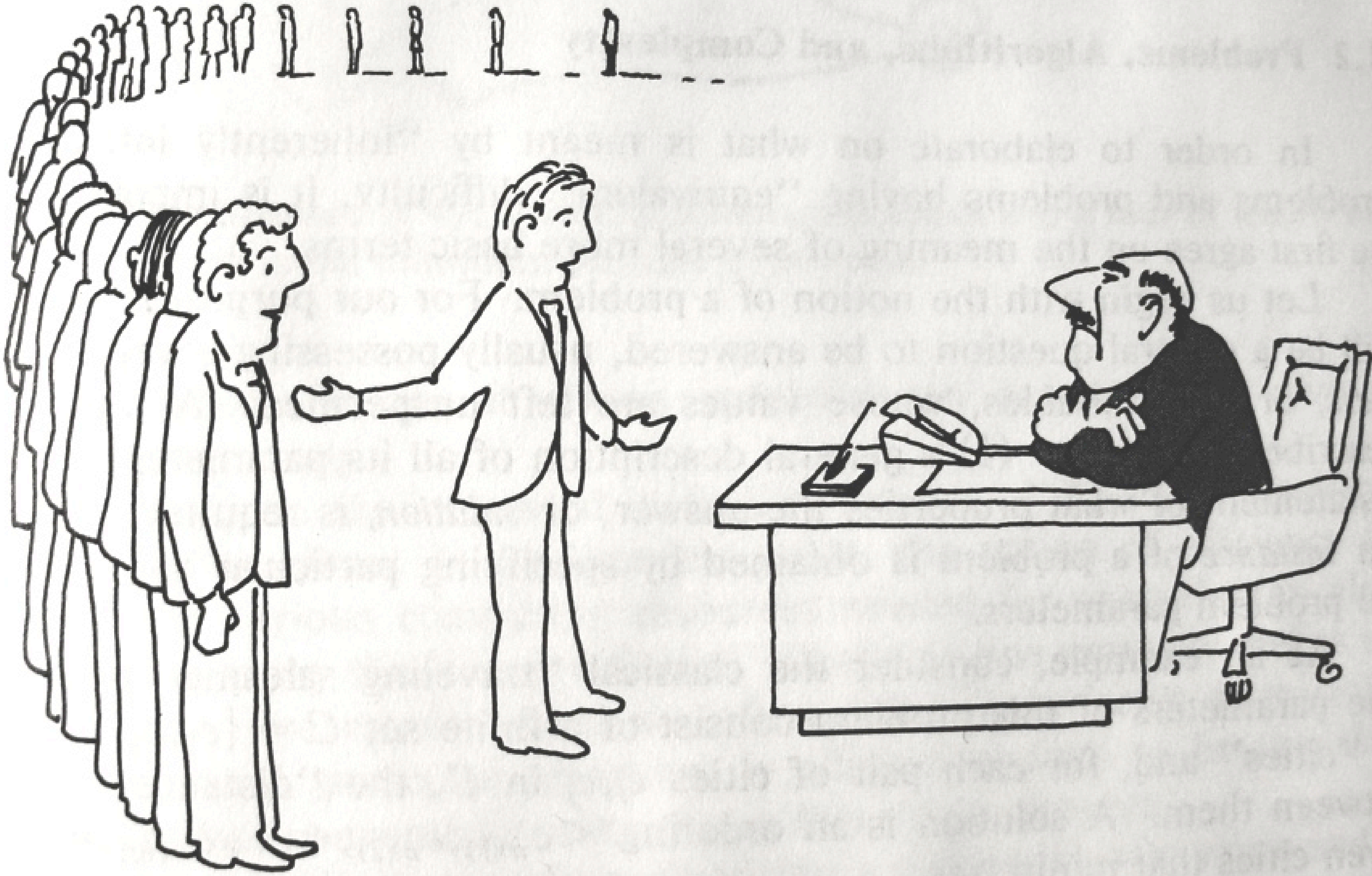
[Garey & Johnson, 1979]



"I can't find an efficient algorithm, I guess I'm just too dumb."



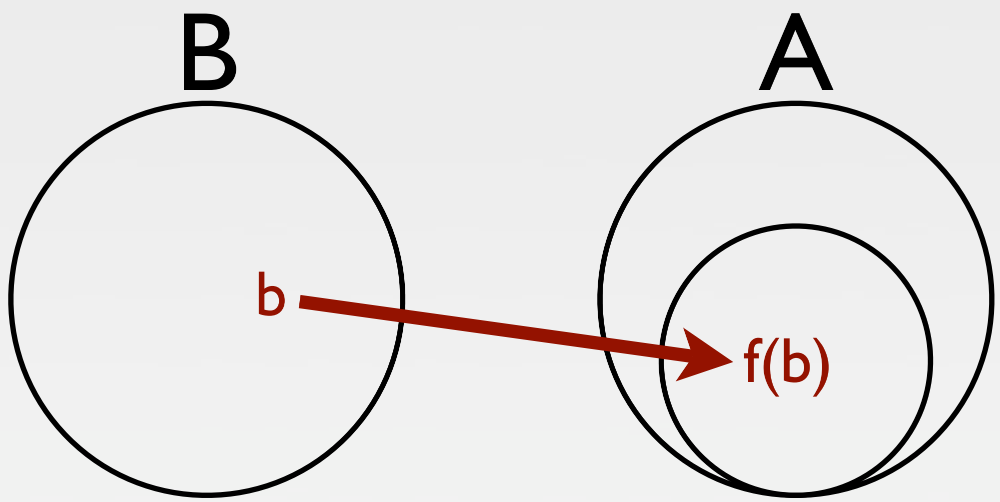
"I can't find an efficient algorithm, because no such algorithm is possible!"



“I can’t find an efficient algorithm, but neither can all these famous people.”

Schwere und einfache Probleme

- Wie zeigt man, dass Problem A
  - in P ist?
    - polynomieller Algorithmus für A
  - NP-schwer ist?
    - Reduktionsbeweis
    - Formuliere A als Entscheidungsproblem.
    - Wähle ein NP-schweres Entscheidungsproblem B.
    - Gebe eine effizient berechenbare Funktion f an, die zu jeder Instanz eines B-Problems b eine Instanz eines A-Problems f(b) liefert, so dass die Lösung von f(b) genau dann *wahr* ist, wenn die von b *wahr* ist.



- SAT (Erfüllbarkeit einer prädikatenlogischen Formel)
  - NP-vollständig, insbesondere für konjunktive Normalform, selbst wenn eine Klausel nur drei Literale enthält (3SAT).

$$(x \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z})$$

Beispiel: Knotenüberdeckung

- Graph G=(V,E) ist ein Tupel aus einer Knotenmenge V und einer Kantenmenge $E \subseteq V x V$.
- Def. (Knotenüberdeckung): Gegeben G=(V,E). $V' \subseteq V$ ist eine Knotenüberdeckung, wenn es für alle $e \in E$ ein $v \in V'$ gibt, so dass $v \in e$.
- Gibt es zu einem gegebenen Graphen eine Knotenüberdeckung bestehend aus höchstens k Knoten?
    - Dieses Problem ist NP-vollständig, d.h. NP-schwer und in NP.
    - in NP: Verifizieren einer KÜ ist einfach.
    - NP-schwer: Beweis mit Hilfe einer Reduktion von 3SAT

3SAT ➠ KÜ Konstruktion

- 3SAT Instanz:
  - n Variablen $V = \{v_1, \dots, v_n\}$
  - m Klauseln $C = \{C_1, \dots, C_m\}$
- KÜ Instanz (durch effizient durchführbare Konstruktion):
  - $\forall v_i : \; V_i = \{v_i, \bar{v}_i\}; \; E_i = \{\{v_i, \bar{v}_i\}\}$
  - $\forall C_j : \; V_j' = \{c_j^1, c_j^2, c_j^3\}; \; E_j' = \{\{c_j^1, c_j^2\}, \{c_j^2, c_j^3\}, \{c_j^1, c_j^3\}\}$
  - $\forall C_j : \; E_j'' = \{\{c_j^1, x\}, \{c_j^2, y\}, \{c_j^3, z\}\}$ wobei $C_j = \{x, y, z\}$
  - $G = (V, E) \quad \text{wobei} \quad V = \left(\bigcup_{i=1}^{n} V_i\right) \cup \left(\bigcup_{j=1}^{m} V_j'\right)$

    $\text{und} \quad E = \left(\bigcup_{i=1}^{n} E_i\right) \cup \left(\bigcup_{j=1}^{m} E_j'\right) \cup \left(\bigcup_{j=1}^{m} E_j''\right)$
  - $k = n + 2m$

3SAT ➠ KÜ Beweis

- Lemma: Jede Knotenüberdeckung enthält mindestens einen Knoten aus jeder V_i Menge und mindestens zwei Knoten aus jeder V_j' Menge.
- Behauptung:
  C ist erfüllbar ⇔ G besitzt KÜ mit höchstens k Knoten
  - Beweis:
    - "*von links nach rechts*": Füge für jede Klausel den Variablenknoten, der zu einem wahren Literal gehört, und die beiden restlichen Literalknoten hinzu. Die resultierende Menge ist eine KÜ mit genau k Knoten.
    - "*von rechts nach links*": Aufgrund des Lemmas kann man aus den Knoten aus E_i, die in K' enthalten sind, eine gültige (d.h. eindeutige) Belegung ablesen. Da K' höchstens k Knoten enthält, befindet sich in jeder Menge E_m" eine Kante, die durch einen Knoten in V_i abgedeckt wird.

Wahlverfahren (1)

- Warum gibt es überhaupt verschiedene Wahlverfahren?
  - Was spricht gegen relative Mehrheit (wahrscheinlich das am weitesten verbreitete Verfahren)?
  - Nehmen wir an es gibt 21 Wähler, von denen jeder die vier möglichen Kandidaten in eine Reihenfolge (abgestuft nach seiner persönlichen Vorliebe) bringen kann.

3	**5**	**7**	**6**
A	A	B	C
B	C	D	B
C	B	C	D
D	D	A	A

  - Kandidat A besitzt eine relative Mehrheit (8 Wähler).
  - Ist A wirklich der beliebteste Kandidat?

Wahlverfahren (2)

- Borda (Slovenien, akademische Institute, Eurovision Song Contest)
  - Die am meisten bevorzugte Alternative eines Wählers erhält k-1 Punkte, die am zweit-meisten bevorzugte Alternative k-2 Punkte, usw. Die Alternative mit den meisten aufsummierten Punkten gewinnt.
- Condorcet-Gewinner
  - Es gewinnt die Alternative, die gegenüber allen anderen Alternativen von einer Mehrheit bevorzugt wird.
- Vorzugswahl (Australien)
  - Die Alternative, die von den wenigsten Wähler am meisten bevorzugt wird, wird gestrichen. Dieser Prozess wird so lange wiederholt bis nur noch eine Alternative übrig ist.
- Relative Mehrheit mit Stichwahl (Frankreich)
  - Die beiden besten Alternativen bezüglich relativer Mehrheit treten in einer Stichwahl gegeneinander an.

Wahlverfahren (3)

- Relative Mehrheit
  - A
- Borda
  - B
- Condorcet-Gewinner
  - C
- Vorzugswahl
  - D
- Relative Mehrheit mit Stichwahl
  - E

33%	16%	3%	8%	18%	22%
A	B	C	C	D	E
B	D	D	E	E	C
C	C	B	B	C	B
D	E	A	D	B	D
E	A	E	A	A	A

Themenübersicht

- Einführung, P und NP, Wahlverfahren
- Präferenzen, Satz von May, Satz von Moulin
- Punkteverfahren und Condorcet-Verfahren
- Fishburns Klassifikation von Condorcet-Verfahren
- Social Choice Correspondences
  - Copeland
  - Good und Schwartz
  - Slater und Banks
  - Uncovered set und Minimal Covering set
- Social Welfare Functions
  - Arrows Unmöglichkeitssatz
  - Dodgson (Lewis Carroll) und Kemeny-Young
  - Ranking Systems und PageRank