Computational Complexity for Computational Social Choice

Summer School on Computational Social Choice

Britta Dorn
Universität Tübingen

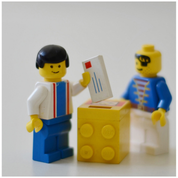
Amsterdam, July 2023

About me

Summer School on Algorithmic Decision Theory, Portugal 2010

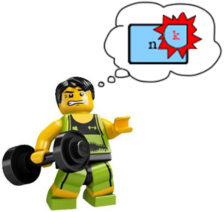
About this lecture

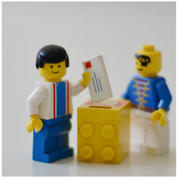
Computational complexity



About this lecture

Parameterized complexity



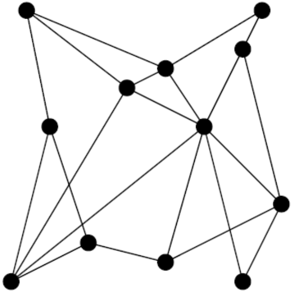


I. Computational Complexity

Computationally hard problems

ICE CREAM SHOP PROBLEM

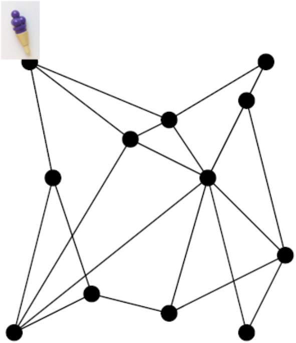
Every street should be covered by an ice cream shop.



Computationally hard problems

ICE CREAM SHOP PROBLEM

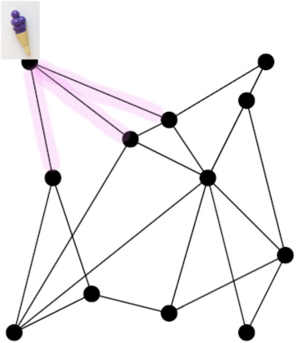
Every street should be covered by an ice cream shop.



Computationally hard problems

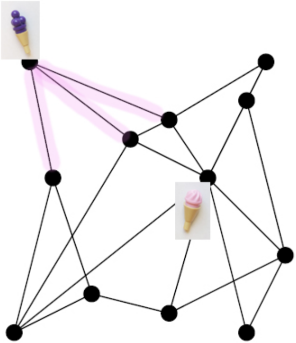
ICE CREAM SHOP PROBLEM

Every street should be covered by an ice cream shop.



Ice Cream shop problem

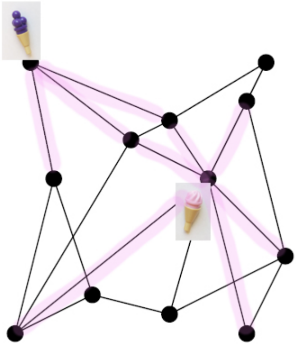
Every street should be covered by an ice cream shop.



Computationally hard problems

ICE CREAM SHOP PROBLEM

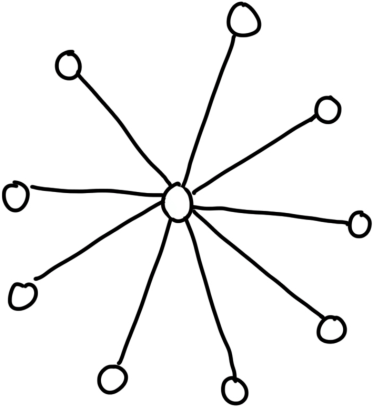
Every street should be covered by an ice cream shop.



Computationally hard problems

Ice cream shop problem

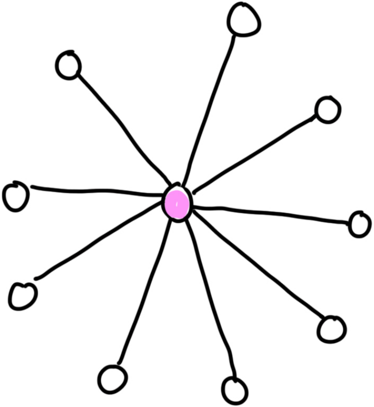
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

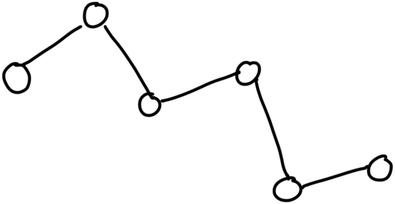
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

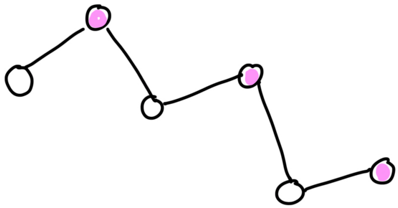
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

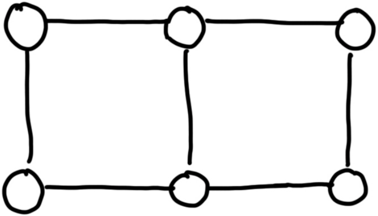
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

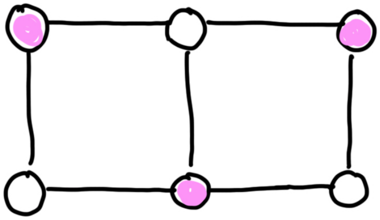
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

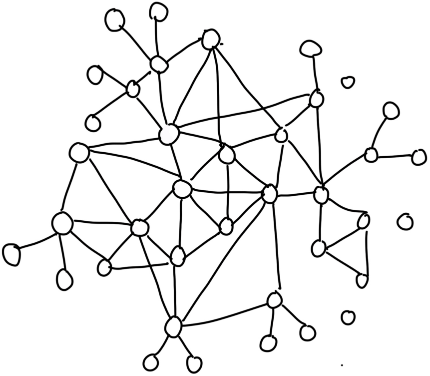
Every street should be covered by an ice cream shop



Computationally hard problems

Ice cream shop problem

Every street should be covered by an ice cream shop



Trying all possible subsets (of size $0, 1, 2, \ldots, n$) of n vertices:
2^n possibilities.

(Too) many possibilities

n	2^n	**running time (10^9 ops./s)**
2	4	0,004 μs
10	1024	1 μs
50	$\sim 10^{15}$	$\sim$ 11,5 days
70	$\sim 10^{21}$	$\sim$ 37.500 years
100	$\sim 10^{30}$	$\sim 3 \cdot 10^{13}$ years

Computationally hard problems

Ice cream shop problem

Every street should be covered by an ice cream shop

is

Vertex Cover

Every edge of the given graph should be covered by a vertex

NP-hard problem

Only algorithms with exponential running time known to solve it exactly.

Computational complexity of a problem

Running time $T(n)$ of an algorithm $\mathcal{A}$: maximal number of computational steps performed by $\mathcal{A}$ on any input of length n, e.g. $2n$, n^3, $n(n+1)$, 2^n, $n!$, …

Problem tractable ("easy")

$T(n)$ is a polynomial of n

Complexity class P

Problem intractable ("hard")

strong evidence that there is no $\mathcal{A}$ that solves it in polynomial time

NP-hard problems

(class NP: given solution can be *checked* in polynomial time)

Running times

Running times (10^9 ops/s)

	polynomial n^2		exponential 2^n	
	n^2	time	2^n	time
$n = 2$	4	0,004 μs	4	0,004 μs
$n = 10$	100	0,1 μs	1024	1 μs
$n = 50$	2 500	2,5 μs	10^{15}	$\sim$ 11,5 days
$n = 100$	10 000	0,01 ms	10^{30}	$\sim 3 \cdot 10^{13}$ years

Computationally hard problems

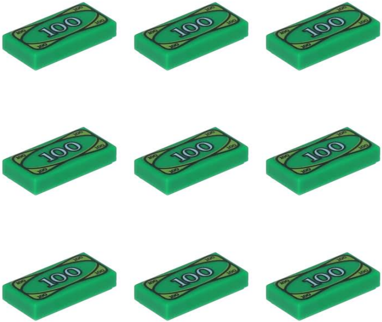
Computational intractability

Strong evidence that NP-hard problems are not solvable in polynomial time.
Essential strategy: brute force — try all possibilities.
Often:

$\mathcal{O}(2^n)$ (try all subsets of the set of size n)
$\mathcal{O}(n!)$ (try all possible orders of the n elements)

Computationally hard problems

P vs. NP problem



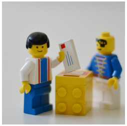
Millennium Prize Problem, US$1 million prize for proving P $\neq$ NP

and US$7 million for proving P $=$ NP (suffices to find a polynomial algorithm for one NP-hard problem)

Computational complexity in COMSOC

Setting: group of agents has to make a common decision, e.g.

- determine a winning alternative



- find a fair allocation

- build coalitions, match persons with tasks, ...

How to solve these problems?

$\rightarrow$ social choice

Computational complexity in COMSOC

How hard is it to solve these problems?

→ computational complexity

- "easy": find efficient algorithms to solve them (P)

  greedy algorithm; flow network; compare with known "easy" problem;

- "hard": understand that it is hard to solve them, no hope for fast solution in general (NP-hardness)

  polynomial time reduction to show that the problem is at least as hard as known "hard" problem

  SAT, Exact Cover with 3-Sets, Clique, ...

Many examples: List of theses in COMSOC ▸ Link

A gentle introduction:

Piotr Faliszewski, Lane A. Hemaspaandra, Edith Hemaspaandra *How Hard Is Bribery in Elections?*, Journal of AI Research, **35, 485–532 (2009).**

Hard problems in COMSOC

Hard problems

Examples:

- Winner determination for certain voting systems is NP-hard
- Determining whether a fair allocation exists can be NP-hard (let alone finding it)

- Dishonest behavior can be NP-hard

  Given: Set of votes, preferred outcome p

  Question: How to change votes in order to obtain p?

  (Strategic voting, bribery, controlling an election)

Hard problems in COMSOC

- Complexity shield!

- Arrow's theorem: Basically no fair voting rules.
  Gibbard-Satterthwaite theorem: basically every voting rule is manipulable.
  $\rightarrow$ Measure the quality of a voting rule in terms of how hard it is to manipulate/bribe/control it!

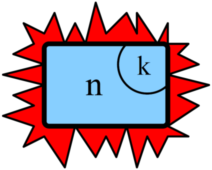
II. Parameterized complexity

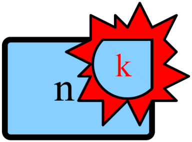
How to deal with NP-hard problems?

- identify easy special cases
- approximation algorithms
- randomization
- parallel computing
- exact exponential algorithms
- heuristics
- parameterized algorithms

Motivation

NP-hard problems: combinatorial explosion of the running time.
BUT in some cases: explosion can be confined to a certain part of the input (parameter, k)

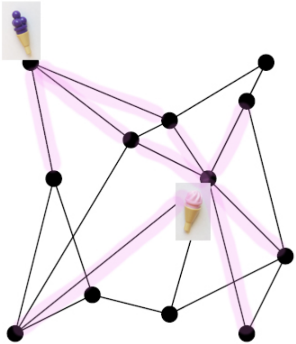




Value of parameter small in certain settings: fast solution possible!

Fixed-parameter tractability

Example: The ICE CREAM SHOP PROBLEM is in FPT for $k = \#$ shops



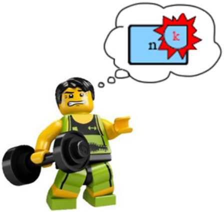
Fixed-parameter tractability

Definition

A problem is called *fixed-parameter tractable* with respect to parameter k, if instances of size n can be solved in

$$f(k) \cdot poly(n) \text{ time} \qquad\qquad (f \text{ – computable function}).$$

complexity class: **FPT**



QUIZ

Which of the following running times are FPT running times with respect to k?

1. $O(2^k \cdot n^{20})$ yes
2. $O(n^2)$ yes
3. $O(2^k \cdot 2^n)$ no
4. $O(1.5^k \cdot n^{1.5})$ yes
5. $O(1.5^k \cdot 1.5^n)$ no
6. $O(n^k)$ no
7. $O(2^{nk})$ no
8. $O(2^k \cdot n^k)$ no
9. $O(n^3 2^{2^{2^k}})$ yes

Fixed-parameter tractability

Parameter: part of the input

Parameters

- often: size of the solution set
- graph problems: degree, treewidth
- nice parameter: distance from tractable instance (with desirable property)

Why is this interesting for COMSOC?

1. Many natural parameters

- # candidates
- # votes
- # candidates with a special property
- # objects
- budget, costs, ...
- variety in voting profile/group
- nice parameter: distance from instance with desirable property
  (single-peakedness of voting profile; stability of matching;
  envy-freeness of allocation)

Why is this interesting for COMSOC?

2. Complexity shield for dishonest behavior:

If hardness is a desirable property:
Make sure the problem **really** is hard!

(justify shield provided by NP-hardness or show it is only an artificial barrier)

$\rightarrow$ Parameterized intractability theory

Ice cream, again

VERTEX COVER

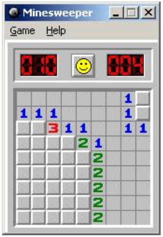
Input:	$G = (V, E)$ undirected graph, $k \in \mathbb{N}$
Question:	Is there a subset $C \subseteq V$ of vertices, $\lvert C\rvert \leq k$, such that every edge in E has at least one endpoint in C?

Data reduction and problem kernels

Data reduction and problem kernels

Idea

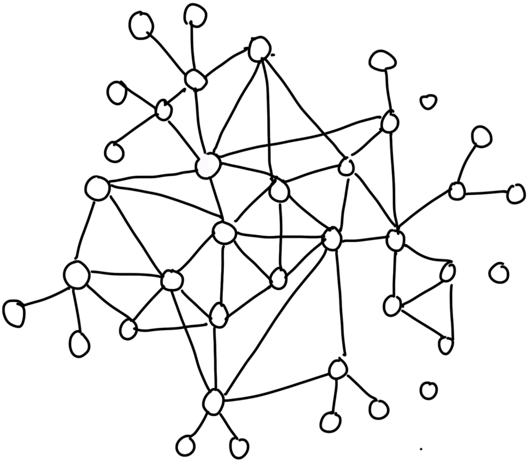
Cut off easy parts of the problem in a polynomial time preprocessing step → hard "kernel"



Goal

End up with a problem kernel whose size depends only on parameter (e.g., size k^2), then brute force on this kernel → **FPT** running time!

Data reduction for VERTEX COVER



Data reduction and problem kernels

Example: VERTEX COVER

Rule 1

Throw away isolated vertices.

Rule 2

Degree-1-vertex: Neighbour is at least as good $\Rightarrow$ Put neighbour in the vertex cover set and delete corresponding vertices and edges.

(OK as long as we are just interested in finding one vertex cover set, not all of them)

Data reduction and problem kernels

Example: VERTEX COVER

Rule 3 (Buss)

Vertices of degree $> k$:
If a vertex v has more than k incident edges, it **has** to be part of the vertex cover set

otherwise, we would have to take all its neighbours, and they are more than k!

- put v in VC
- delete v and all its incident edges from G
- set $k' := k - 1$

Problem kernel for VERTEX COVER

Theorem (Buss)

After applying Rules 1,2,3 exhaustively, the remaining graph either consists of $\leq k^2 + k$ vertices and k^2 edges, or a vertex cover of size k cannot exist for the original graph.

Example from COMSOC: Kemeny ranking

Given: individual rankings over set of candidates:

v_1: $a > b > c > d$

v_2: $b > a > c > d$

v_3: $b > c > a > d$

Wanted: consensus ranking (minimizing Kendall-Tau-distance)

Count inversions (dirty pairs):

v_1 and v_2: inversion $a \leftrightarrow b$, distance is 1

v_1 and v_3: inversions $a \leftrightarrow b$, $a \leftrightarrow c$, distance is 2

Consensus ranking

Given rankings and $k \in \mathbb{N}$, is there a ranking with total distance $\leq k$ from all rankings?

(Here: yes for $k = 2$: ranking $b > a > c > d$)

NP-hard problem

Example from COMSOC: Kemeny ranking

Given: individual rankings:
$v_1: a > b > c > d$
$v_2: b > a > c > d$
$v_1: b > c > a > d$

Is there a consensus ranking with total distance $\leq k$ from all rankings?

Rule 1

Delete all candidates that are in no dirty pair.

here: candidate d.

After Rule 1: If there is a such a consensus, there must be $\leq 2k$ candidates.

Only dirty candidates remaining in reduced instance. More than $2k$ candidates $\Rightarrow$ more than k dirty pairs.

For each pair, two possibilities to position candidate in consensus ranking, adding one inversion for each possibility,

but consensus has $\leq k$ inversions.

Example from COMSOC: Kemeny ranking

Is there a consensus ranking with distance $\leq k$ from all rankings?

Rule 2

If there are more than k rankings identical (ranking r):
Choose r as consensus and check whether it has a total of $\leq k$ inversions with all other rankings.

If we do not choose r, then there is at least one inversion with each of the k identical rankings, distance $\geq k$

After Rule 1 and 2: At most $2k$ votes in a yes-instance.

Between two distinct rankings: distance ≥ 1. At most k copies of each ranking $\Rightarrow$ not more than $2k$ votes if consensus has total distance $\leq k$.

Example from COMSOC: Kemeny ranking

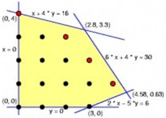
Result
After applying Rules 1 and 2 exhaustively, the remaining instance either consists of $\leq 2k$ candidates and $\leq 2k$ votes, or a consensus of distance $\leq k$ cannot exist.

Nadja Betzler, Michael R. Fellows, Jiong Guo, Rolf Niedermeier, Frances A. Rosamond, Fixed-parameter algorithms for Kemeny rankings, Theoretical Computer Science **410**:45, 4554–4570, 2009.

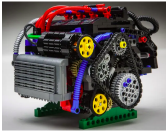
More techniques to prove fixed-parameter tractability

- Depth-bounded search trees

- Color-coding



- Integer Linear Programming



- Iterative compression

Parameterized intractability

- FPT reduction
- Parameterized complexity classes

Advanced techniques

- Approximation and parameterized algorithms
- Parameterized complexity for problems beyond NP

Some references

Parameterized complexity, general

M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*, Springer, 2015.

R.G. Downey and M.R. Fellows. *Parameterized Complexity*, Springer, 1999.

R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*, Springer, 2013.

J. Flum and M. Grohe. *Parameterized Complexity Theory*, Springer, 2006.

R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.

R. de Haan. *Parameterized Complexity in the Polynomial Hierarchy*, PhD thesis, Technische Universität Wien, 2016b.

Parameterized complexity in COMSOC, overviews/surveys

N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. *Studies in computational aspects of voting—a parameterized complexity perspective*. In H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors, The Multivariate Algorithmic Revolution and Beyond, 18–363, 2012.

R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. *Parameterized algorithmics for computational social choice: Nine research challenges*, Tsinghua Science and Technology, 19(4):358–373, 2014

B. Dorn, I. Schlotter. *Having a Hard Time? Explore Parameterized Complexity!* In U. Endriss, editor, Trends in Computational Social Choice, AI Access, 2017.

P. Faliszewski and R. Niedermeier. *Parameterization in computational social choice*, In M.-Y. Kao, editor, Encyclopedia of Algorithms. Springer, 2015.

C. Lindner and J. Rothe. *Fixed-parameter tractability and parameterized complexity applied to problems from computational social choice*, In A. Holder, editor, Mathematical Programming Glossary. INFORMS Computing Society, 2008.

Summer School on Algorithmic Decision Theory, Portugal 2010

Six years later

A Discrete and Bounded Envy-Free Cake Cutting Protocol for Four Agents

Haris Aziz Simon Mackenzie
Data61 and UNSW
Sydney, Australia
{haris.aziz, simon.mackenzie}@data61.csiro.au

ABSTRACT

We consider the well-studied cake cutting problem in which the goal is to identify an envy-free allocation based on a minimal number of queries from the agents. The problem has attracted considerable attention within various branches of computer science, mathematics, and economics. Although, the elegant Selfridge-Conway envy-free protocol for three agents has been known since 1960, it has been a major open problem to obtain a bounded envy-free protocol for more than three agents. The problem has been termed the central open problem in cake cutting. We solve this problem by proposing a discrete and bounded envy-free protocol for four agents.

problem of fairly dividing the cake is a fundamental one within the area of fair division and multiagent resource allocation [6, 17, 26, 29, 33, 35, 36].

Formally speaking, a cake is represented by an interval [0, 1] and each of the n agents has a value function over pieces of the cake that specifies how much that agent values a particular subinterval. The main aim is to divide the cake fairly. In particular, an allocation should be envy-free so that no agent prefers to take another agent's allocation instead of his own allocation. Although an envy-free allocation is guaranteed to exist even with $n-1$ cuts [35][1], *finding* an envy-free allocation is a challenging problem which has been termed "one of the most important open problems in 20th century mathematics" by Garfunkel [16].

Summer school outcomes

6. ACKNOWLEDGMENTS

Data61 is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. Part of the research was carried out when the authors were visiting LAMSADE, University Paris Dauphine in May 2015. The authors thank Rediet Abebe, Steven Brams, Simina Brânzei, Ioannis Caragiannis, Katarína Cechlárová, Serge Gaspers, David Kurokawa, and Ariel Procaccia for useful comments. They also thank the reviewers of STOC 2016 for suggestions to improve the presentation. Haris Aziz thanks Ulle Endriss for introducing the subject to him at the COST-ADT Doctoral School on Computational Social Choice, Estoril, 2010.

Thank you!