Benefit of Promises and Discrete Advice

Klaus Ambos-Spies, Ulrike Brandt, Martin Ziegler
(Univ. Heidelberg) (TU Darmstadt)

Enrichment in Recursive Analysis: Computing with Discrete Advice

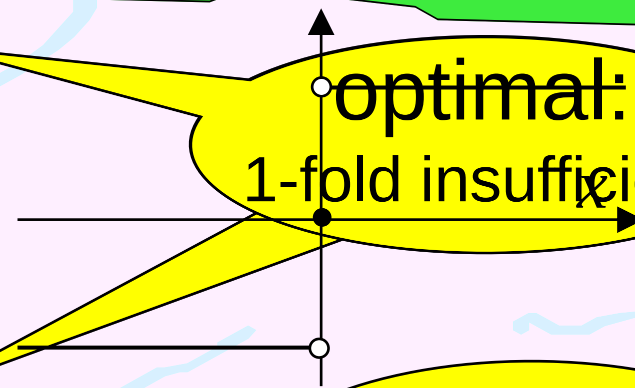
2-fold advice "x=0?" makes sign uniformly recursive.

discontinuous floor function:

$\lfloor x \rfloor$ itself as advice is unbounded.

2-fold advice: "$x \in \mathbb{Z}$?" suffices.

Alternative 2-fold advice: parity($\lfloor x \rfloor$)

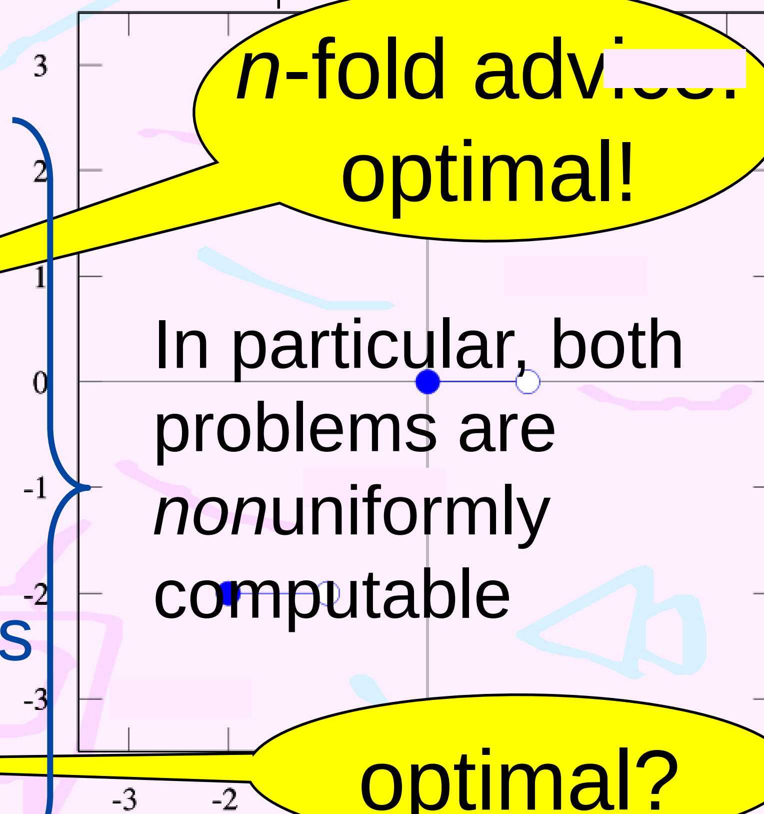


Given singular $A \in \mathbb{R}^{d \times d}$, nontriv.
solve linear equation $A \cdot \underline{x} = 0$

[Brattka&Z.04]: knowing rank(A)$\in${0,1,…,d-1} suffices for computing a basis of kernel(A)

Given symmetric $A \in \mathbb{R}^{d \times d}$,
compute eigenvector/eigenbasis

[Brattka&Z.04]: knowing Card(σ(A))$\in${1,…,d} suffices



optimal?

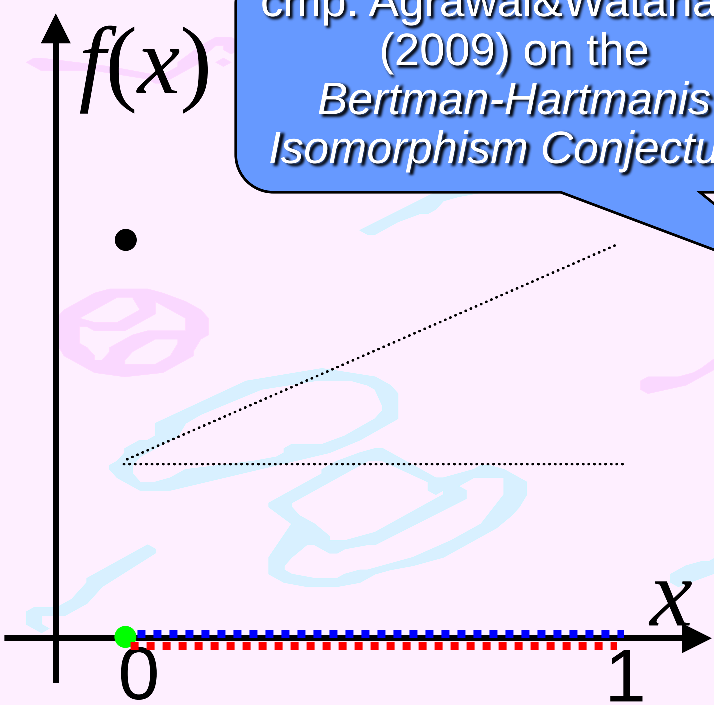
Computing with Advice Formalized

Multi- Function $f{:}X \rightrightarrows Y$ sensible also for $X=\mathbb{N}$ or $\{0,1\}^*$

K-fold advice: provide to each $x \in X$ some $k(x) \in \{1,\ldots K\}$

Induces partition $\Delta = \{\, k^{-1}[1], k^{-1}[2], \ldots, k^{-1}[K] \,\}$ of X.

Call f computable with advice Δ in polyn. time if $f|_D$ computable in polyn. time $\forall D \in \Delta$



cmp. Agrawal&Watanabe (2009) on the *Bertman-Hartmanis Isomorphism Conjecture*

Skew to classical nonuniform (=circuit) complexity classes:

$\mathcal{P}$/poly non-constant size

$\mathcal{P}$/const advice may depend only on length of x

Every $f{:}X \to \{1,\ldots K\}$ is trivially computable with K-fold advice.

Real Function Complexity with/-out 2-fold Advice

time hierarchy theorem (diagonalization), Hartmanis&Stearns'65

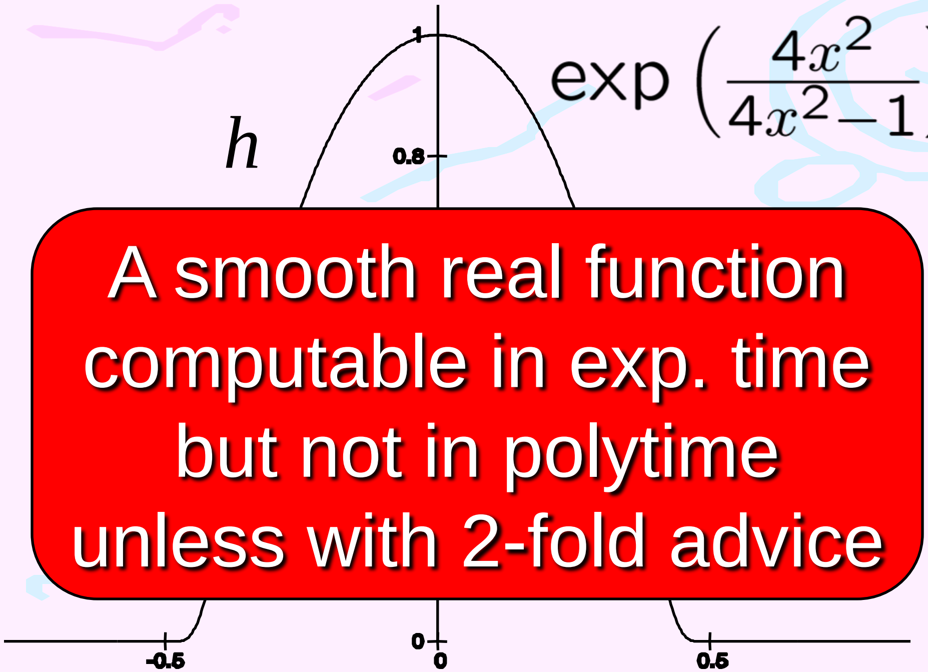
Let $L\subseteq\mathbb{N}$ be decidable in exp. but not in polyn. time

2-fold advice $(L,\mathbb{N}\setminus L)$ renders it polytime decidable.

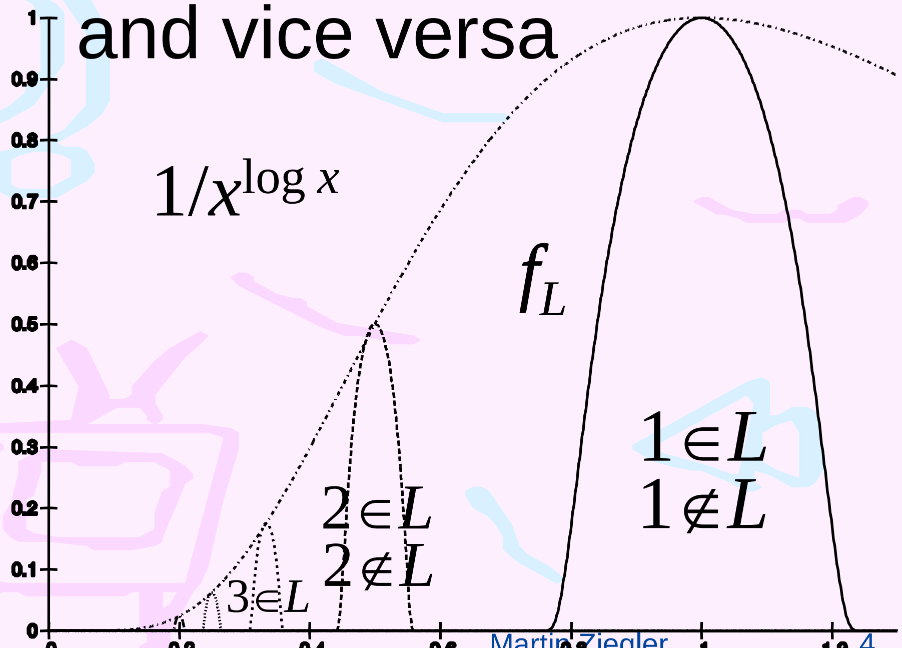
$h(x):=\exp(-4x^2/(1-4x^2))$ for $|x|\leq½$, $h(x):=0$ for $|x|\geq½$

$f_L(x) := \sum_{m\in L} h(m\cdot x-1)/m^{\log m}$

polytime computable with 2-fold advice and vice versa



A smooth real function computable in exp. time but not in polytime unless with 2-fold advice



Generalized Decision Problems

"Given a Boolean formula with ≤1 sat. assignment does it admit a satisfying assignment?"

→not a language (=decision problem)

→$\mathcal{NP}$-complete

Decision problem: $L\subseteq\mathbb{N}$ or $L\subseteq\{0,1\}^*$

→ "Given $x\in\mathbb{N}$, does it hold $x\in L$ or $x\notin L$?"

Oded Goldreich
LNCS 3895
(2005)

Promise problem: (A,B) with $A\cap B=\emptyset$

→ "Given $x\in A\cup B$, does it hold $x\in A$ or $x\in B$?"

Def: Classification problem is a tuple $C=(C_1,\ldots C_d)$. An algorithm solves C if, on input $x\in\bigcup C$, it outputs $j\in\{1,\ldots d\}$ with $x\in C_j$.

not neces. disjoint→multival

algorithm may behave arbitrarily on inputs $x\notin A\cup B$

Main Result: Discrete

Fix $J \in \mathbb{N}$. There is a classfication problem $(C_1, \ldots C_J)$

- algorithmical. solvable time a tower of height J-1 but not one of height J-2
- with 2-fold advice time a tower of height J-2 but not one of height J-3
- … …
- with $(J\text{-}2)$-fold advice in doubly expon. time but not in expon. time
- with $(J\text{-}1)$-fold advice in exponential time but not in polyn. time
- with J-fold advice in polynomial time (triv.ly)

Main Result: Smooth

Fix $J\in\mathbb{N}$. There exists a C^∞ function $f_J:[0;1]\to[0;1]$

Fix $J\in\mathbb{N}$. There is a classification problem $(C_1,\ldots,C_J)$

- computable:
- algorithmical. solvable — time a tower of height J-1 but not one of height J-2
- with 2-fold advice — time a tower of height J-2 but not one of height J-3
- …

$$f\,(x) := \textstyle\sum_{j=1\ldots J}\; j\cdot\sum_{m\in C_j}\; h(m\cdot x\text{-}1)/m^{\log m}$$

- with (J-2)-fold advice — in doubly expon. time but not in expon. time
- with (J-1)-fold advice — in exponential time but not in polyn. time
- with J-fold advice — in polynomial time

Proof (Sketch)

Lemma: a) To $d,J\in\mathbb{N}$ there exists a total classification problem $\mathcal{C}^{d,J}=(C_1,\ldots C_d)$ solvable in time a tower of height J but not a tower of height J-1, even with (d-1)-fold advice.

b) Fix classification problems $\mathcal{C}=(C_1,\ldots C_d)$, $\mathcal{B}=(B_1,\ldots B_d)$.
Complexity of $\mathcal{B}\oplus\mathcal{C} := \big((0\circ B_1)\cup(1\circ C_1),\ldots,(0\circ B_d)\cup(1\circ C_d) \big)$ is the maximum of $\mathcal{B}$ and $\mathcal{C}$. $(\mathcal{A}\oplus\mathcal{B})\oplus\mathcal{C} \equiv \mathcal{A}\oplus(\mathcal{B}\oplus\mathcal{C})$

Proof: a) diagonalization; b) immediate reduction.

Finally consider $\mathcal{C}^{2,J-1} \oplus \mathcal{C}^{3,J-2} \oplus \mathcal{C}^{4,J-3} \oplus \ldots \oplus \mathcal{C}^{J-1,2} \oplus \mathcal{C}^{J,1}$:
solvable in time a tower of height J-1, but not of height J-2
with 2-fold advice a tower of height J-2, but not height J-3,
with 3-fold advice a tower of height J-3, but not height J-4

A more natural example (?)

The functional $\max : C[-1;1] \to \mathbb{R}$, $f \to \max_x f$

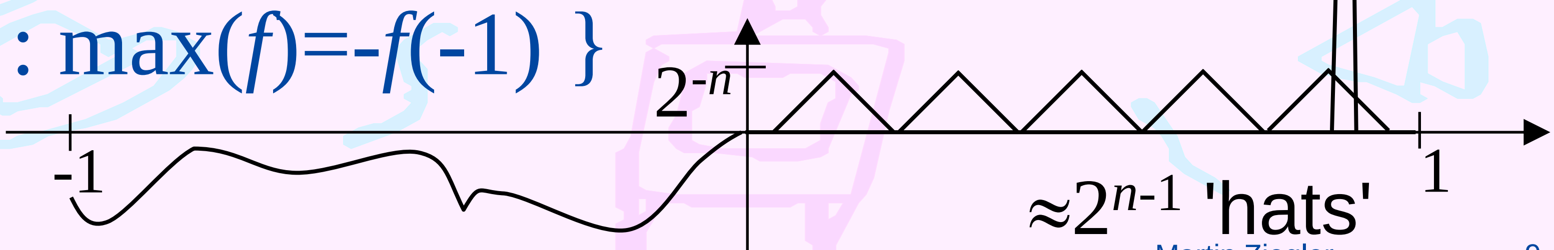
Friedman &Ko 1982

- can map polytime $f \in C^\infty[-1;1]$ to $\mathcal{NP}_1$-hard $\max(f)$
- uniformly requires expon.time, even on $1\text{-Lip}[0;1]$
- adversary argument, adapted from IBC (Traub…)
- $\max$ is uniformly easy when restricted to $\mathcal{K}$ or $\mathcal{L}$
- but not on $\mathcal{K} \cup \mathcal{L}$:

2-fold advice drops complexity from exponent. to polynom.

$\mathcal{K} := \{ f : f(x)=0 \text{ for } x \geq 0,\ f(x) \leq 0 \text{ for } x \leq 0 \}$

$\mathcal{L} := \{ f : \max(f) = -f(-1) \}$



Conclusion

In practice, (real number inputs often exhibit some structure (e.g. band-3 matrix from FEM method)

Without exploiting such information, or merely detecting it, (i.e. *general purpose*) algorithms are often inefficient;

the problem may even be discontinuous/uncomputable.

"**Discrete advice**" formalizes such additional information;

its necessity for real computability is well-known in TTE.

"How much" discrete advice is necessary and sufficient to render a real number problem <u>computable</u>/continuous?

$\rightarrow$ <u>topological</u> complexity theory

Conclusion

Today: Discrete advice and computational complexity
- A real function, computable but of high complexity
- which gradually drops with increasing discrete advice.

Technique: encode discrete classification problem as natural generalization of standard promise problems.

$\to$ Artificial problem (diagonalization); more natural:

Maximization functional on certain subspace of $C[0;1]$
- uniformly computable in exponential time
- drops to polytime with 2-fold advice.

$\to$ artificial domain: How about *really* natural examples?

"How much" discrete advice is necessary and sufficient to render a real number problem computable/continuous?

$\to$ topological complexity theory