

Proof Complexity of Search Problems

Pavel Pudlák

*Mathematical Institute, Academy of Sciences, Prague
and
Charles University, Prague*

Barriers in Complexity, Princeton, 2009

Overview

- total polynomial search problems—definition and reductions
- examples of natural classes
- relativized separations
- formalization of total search problems in first order theories
- a characterization of provably total search problems in fragments of Bounded Arithmetic

Definition

A total **NP** search problem (**TFNP**) is given by a relation R such that

- 1 $R(x, y) \in \mathbf{P}$;
- 2 there is a polynomial p such that $R(x, y)$ implies $|y| \leq p(|x|)$;
- 3 $\forall x \exists y R(x, y)$.

The problem is: *given input x , find y such that $R(x, y)$.*

Why they are important:

- classification of functional problems;
- characterization of $\forall \Sigma_1$ theorems of bounded arithmetical theories.

Basic facts

1. $\mathbf{P} = \mathbf{NP}$ implies that every **TFNP** can be solved in polynomial time.
2. If every **TFNP** can be solved in polynomial time, then $\mathbf{NP} \cap \mathbf{coNP} = \mathbf{P}$.

Reductions between TFNP search problems

Let S_i be a search problem determined by $R_i(x, y)$. Then S_1 is

- 1 *Turing reducible* to S_2 if there exists a polynomial time algorithm that queries S_2 for solutions and solves S_1 ;
- 2 *many-one reducible* to S_2 if there exist polynomial time computable functions f and g such that given x , f computes some string $f(x) = x'$ such that if $R_2(x', y')$ for some y' , then $R_1(x, g(x, y'))$ (\Leftrightarrow Turing reducible using a single query).

$$\begin{array}{ccc} R_1(& x & y &) \\ & \downarrow f & \uparrow g & \\ R_2(& x' & y' &) \end{array}$$

Conjecture

There is no complete **TFNP** search problem.

Classes of natural search problems

Johnson, Papadimitriou, Yannakakis 1988: several classes defined by combinatorial properties.

Definition

A *class* of search problems is a set of search problems closed under polynomial time reductions.

FP – search problems solvable in polynomial time.

PPAD – *Polynomial Parity Argument Directed* is the class of problems polynomially reducible to:

1. A directed graph on $\{0, 1\}^n$ with indegrees and outdegrees ≤ 1 is given by: *two polynomial time computable functions, one for the predecessor and one for the successor of a vertex.*
2. $\vec{0}$ is a source.
3. Find a sink in the graph.

PPAD contains a number of important search problems:

- Brouwer's Fixedpoint Theorem
- Sperner Lemma
- Nash Equilibrium

PPAD contains a number of important search problems:

- Brouwer's Fixedpoint Theorem
- Sperner Lemma
- Nash Equilibrium

An application

Theorem (Daskalakis, Goldberg, Papadimitriou, Chen, Deng, 2005)

*Finding a Nash Equilibrium (with exponential precision) is a **PPAD** complete search problem.*

Computing the minimax (i.e., the equilibrium in a zero-sum game) is an LP problem, hence can be done in *polynomial time*.

Can a Nash equilibrium in general games be found in polynomial time?

We conjecture NO. We believe that **PPAD** are not solvable in polynomial time, because there exists an oracle A such that $\mathbf{PPAD}^A \not\subseteq \mathbf{FP}^A$ and Nash Equilibrium is complete in **PPAD**.

FACTORING – search problems reducible to *FACTORING*.

FACTORING is the search problem defined by

$P(x, y) \Leftrightarrow x$ is a prime, or y is a proper factor of x

FACTORING – search problems reducible to *FACTORING*.

FACTORING is the search problem defined by

$P(x, y) \Leftrightarrow x$ is a prime, or y is a proper factor of x

Problem

Find a “combinatorial” definition of this class, ie., in terms of polynomial time computable structures.

If we had such a definition, we could relativize the class and possibly show that it is not in the relativized polynomial time.

PLS – Polynomial Local Search

1. The problem is given by polynomial time computable functions $v(x, y)$ (the *cost function*) and $h(x, y)$ (the *neighborhood functions*) such that $v(x, y), h(x, y) \leq x$.
2. The task is: for a given a , find $b \leq a$ such that

$$v(a, b) \geq v(a, h(a, b)).$$

The natural exponential time algorithm to find a solution is:

```
b:=0  
do b := h(a, b) while v(a, b) < v(a, h(a, b))
```

PLS – Polynomial Local Search

1. The problem is given by polynomial time computable functions $v(x, y)$ (the *cost function*) and $h(x, y)$ (the *neighborhood functions*) such that $v(x, y), h(x, y) \leq x$.
2. The task is: for a given a , find $b \leq a$ such that

$$v(a, b) \geq v(a, h(a, b)).$$

The natural exponential time algorithm to find a solution is:

```
b:=0
do b := h(a, b) while v(a, b) < v(a, h(a, b))
```

PPP – Polynomial Pigeonhole Principle

This class can be defined as the search problems reducible to *PHP*, *Pigeon Hole Principle*, which is defined by:

1. Given a polynomial time computable function $f : [0, N + 1] \rightarrow [0, N]$,
2. find $x, y \in [0, N + 1]$, $x \neq y$ such that $f(x) = f(y)$.

The only algorithm we know is to search all pairs (x, y) .

Relativizations

If a search problem is defined by a polynomial time structure, then there is a natural way to relativize it to an oracle A —replace ‘*polynomial time computable*’ by ‘*polynomial time computable using oracle A* ’.

Formally, this is defined using type 2 search problems.

If the problem is given by a first order structure $(X; R_1, \dots, R_k, F_1, \dots, F_l)$ we can take just the relations R_i and the functions F_j as oracles.

Thus reductions are (type 2) *polynomial time functionals*. Such reductions are called *generic*.

It is easy to construct oracles that separate all classes mentioned above, except for **FP** and **FACTORING**, from **FP**.

Theorem (Riis 1993, see also Krajíček 1995, Morioka 2001)

Let a search problem S be defined by an existential sentence $\exists y\phi$ in the language $(R_1, \dots, R_k, F_1, \dots, F_l)$ where all R_i and F_j are generic relations and functions (thus we do not allow arithmetical operations on integers etc.).

If $\neg\exists y\phi$ has an infinite model, then $S^A \notin \text{PLS}^A$,

for some oracle A .

Example. $\text{PPP}^A \not\subseteq \text{PLS}^A$

— there exists a model in which PHP fails

Let P_i , $i = 1, 2$ be two search problems defined by existential formulas. Let $\{\alpha_n\}$, resp. $\{\beta_n\}$ be a sequence of propositions (tautologies) expressing the totality of problem P_1 , resp. P_2 .

Theorem (Buresh-Oppenheim, Morioka)

If there exists a generic reduction of P_1 to P_2 , then tautologies $\{\alpha_n\}$ have polynomial size bounded depth propositional proofs from tautologies $\{\beta_n\}$.

Let P_i , $i = 1, 2$ be two search problems defined by existential formulas. Let $\{\alpha_n\}$, resp. $\{\beta_n\}$ be a sequence of propositions (tautologies) expressing the totality of problem P_1 , resp. P_2 .

Theorem (Buresh-Oppenheim, Morioka)

If there exists a generic reduction of P_1 to P_2 , then tautologies $\{\alpha_n\}$ have polynomial size bounded depth propositional proofs from tautologies $\{\beta_n\}$.

Many oracle separations were proved by Beame, Cook, Edmonds, Impagliazzo, Pitassi, Buresh-Oppenheim, Morioka, . . .

Search problems in first order theories

Recall that a search problem is given by a relation R such that

- 1 $R(x, y) \in \mathbf{P}$;
- 2 there is a polynomial p such that $R(x, y)$ implies $|y| \leq p(|x|)$;
- 3 $\forall x \exists y R(x, y)$.

Question: What theory do we need to prove $\forall x \exists y R(x, y)$?

Search problems in first order theories

Recall that a search problem is given by a relation R such that

- 1 $R(x, y) \in \mathbf{P}$;
- 2 there is a polynomial p such that $R(x, y)$ implies $|y| \leq p(|x|)$;
- 3 $\forall x \exists y R(x, y)$.

Question: What theory do we need to prove $\forall x \exists y R(x, y)$?

Conjecture

*There exists no r.e. true theory T such that every **TFNP** problem is reducible to a search problem that is *provably total in T* .*

Slightly stronger version:

Conjecture

For every r.e. consistent theory T , there exists a total polynomial search problem S which is strictly stronger than all search problems provably total in T .

If true, it would show that **incompleteness can be caused by complexity**.

Slightly stronger version:

Conjecture

For every r.e. consistent theory T , there exists a total polynomial search problem S which is strictly stronger than all search problems provably total in T .

If true, it would show that **incompleteness can be caused by complexity**.

Conjecture

There exists no complete problem among total polynomial search problems, that is, no problem to which total polynomial search problems are reducible.

This three conjectures are equivalent. Hint:

Lemma

*For every r.e. consistent theory T , there exists a **TFNP** problem S_T , such that every **TFNP** problem probably total in T is polynomially reducible to it.*

Feasible incompleteness conjectures

Example:

Theorem (well-known)

For every r.e. consistent theory T , there exists a recursive function that grows faster than all recursive functions that are provably total in T .

fast growth \mapsto high complexity

Conjecture (feasible version of the theorem)

For every r.e. consistent theory T , there exists a total polynomial search problem S which is strictly stronger than all search problems provably total in T .

A specialization of this problem:

Problem

- Given a total search problem S , find the weakest *natural* theory T in which it is provably total.
- Given a theory T , find the strongest *natural* total search problem that is provably total in T .

A specialization of this problem:

Problem

- Given a total search problem S , find the weakest *natural* theory T in which it is provably total.
- Given a theory T , find the strongest *natural* total search problem that is provably total in T .

Our “natural” theories will be **theories associated with complexity classes**:

\mathcal{C} complexity class $\leftrightarrow \Theta_{\mathcal{C}}$ first order theory

- suitable language L
- class of formulas Γ that characterizes the class \mathcal{C}
- basic axioms (fixing the interpretation of primitive notions)
- **induction for formulas of Γ**

Informally: $\Theta_{\mathcal{C}}$ is “induction for \mathcal{C} ”.

Problem

$$\Theta_P = \Theta_{NP}?$$

Problem

$$\Theta_{\mathbf{P}} = \Theta_{\mathbf{NP}}?$$

We only know:

Theorem (Krajíček, P., Takeuti 1991)

If $\mathbf{NP} \not\subseteq \mathbf{P}/\text{poly}$ then $\Theta_{\mathbf{P}} \neq \Theta_{\mathbf{NP}}$.

complexity class	theory	total search problem	
P	Θ_P	FP	Buss 1986
NP	Θ_{NP}		

complexity class	theory	total search problem	
P	Θ_P	FP	Buss 1986
NP	Θ_{NP}	PLS	Buss-Krajíček 1994
Σ_2^P	$\Theta_{\Sigma_2^P}$	CPLS	Krajíček-Skalley-Thapen 2006
...	
Σ_k^P	$\Theta_{\Sigma_k^P}$	GI_k	Skalley-Thapen 2008
...	

complexity class	theory	total search problem	
P	Θ_P	FP	Buss 1986
NP	Θ_{NP}	PLS	Buss-Krajíček 1994
Σ_2^P	$\Theta_{\Sigma_2^P}$	CPLS	Krajíček-Skalley-Thapen 2006
...	
Σ_k^P	$\Theta_{\Sigma_k^P}$	GI_k	Skalley-Thapen 2008
...	

Problem

Justify the correspondence $P \leftrightarrow FP$, $NP \leftrightarrow PLS$, $\Sigma_2^P \leftrightarrow CPLS$, etc., without using logic!

Bounded Arithmetic [Buss 1986]

- For every $k \geq 1$, a class of formulas Σ_k^b that define sets of the complexity class Σ_k^P .
- Theories T_2^k (formalizations of $\Theta_{\Sigma_k^P}$) based on **induction** for Σ_k^b formulas.
- Theories T_2^k can also be axiomatized by **the least number principle** for Σ_k^b formulas.

I will show that T_2^k can also be axiomatized by postulating the existence of **k -times alternating minima and maxima**.

Notation.

x_0 will be a parameter;

$v(x_0, x_1, \dots, x_k)$ a polynomial time computable function with values $\leq x_0$;

quantification of x_i , for $i = 1, \dots, k$, will always be over the domain $0 \leq x_i \leq x_0$,
the parameter x_0 will often be omitted.

Notation.

x_0 will be a parameter;

$v(x_0, x_1, \dots, x_k)$ a polynomial time computable function with values $\leq x_0$;

quantification of x_i , for $i = 1, \dots, k$, will always be over the domain $0 \leq x_i \leq x_0$, the parameter x_0 will often be omitted.

We want to characterize

$$W = \min\{w; \exists x_1 \forall x_2 \exists x_3 \dots v(x_1, \dots, x_k) \leq w\}.$$

Consider the game in which players \exists and \forall alternate in choosing x_1, \dots, x_k . Player \exists wants to **minimize** the value v , player \forall wants to **maximize** the value v , (because when they finish, \exists loses (resp. \forall gains) $v(x_1, \dots, x_k)$ points).

If they play optimally, the value is W . Thus

$$\min_{x_1} \max_{x_2} \min_{x_3} \dots v(x_1, \dots, x_k) = \min\{w; \exists x_1 \forall x_2 \exists x_3 \dots v(x_1, \dots, x_k) \leq w\}.$$

Theorem

The existence of the minimum

$$\min\{w; \exists x_1 \forall x_2 \exists x_3 \dots v(x_1, \dots, x_k) \leq w\}$$

is equivalent to the sentence

$$\exists x_1 \forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \dots (v(x_1, \dots, x_k) \leq v(y_1, \dots, y_k)). \quad (1)$$

$$\begin{array}{ccccccc} \exists x_1 & & \forall x_2 & \rightarrow & \exists x_3 & & \dots & v(x_1, \dots, x_k) \\ \downarrow & & \uparrow & & \downarrow & & & \leq \\ \forall y_1 & \rightarrow & \exists y_2 & & \forall y_3 & \rightarrow & \dots & v(y_1, \dots, y_k) \end{array}$$

Proof-idea.

1. Show that the existence of the minimum is equivalent to the following sentence

$$\exists w [\exists x_1 \forall x_2 \exists x_3 \dots (v(x_1, \dots, x_k) \leq w) \wedge \forall y_1 \exists y_2 \forall y_3 \dots (w \leq v(y_1, \dots, y_k))].$$

2. Transform the formula into the prenex form

$$\exists w \exists x_1 \forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \dots (v(x_1, \dots, x_k) \leq w \wedge w \leq v(y_1, \dots, y_k)).$$

3. Eliminate w .

Corollary

T_2^k can be axiomatized by the sentence

$$\forall x_0 \exists x_1 \leq x_0 \forall y_1 \leq x_0 \exists y_2 \leq x_0 \forall x_2 \leq x_0 \exists x_3 \leq x_0 \forall y_3 \leq x_0 \dots$$

$$v(x_0, x_1, \dots, x_k) \leq v(x_0, y_1, \dots, y_k),$$

where v denotes a polynomial time computable function.

Game-theoretical interpretation of the sentence (1)

$$\begin{array}{ccccccc} \exists x_1 & & \forall x_2 & \rightarrow & \exists x_3 & & \dots & v(x_1, \dots, x_k) \\ \downarrow & & \uparrow & & \downarrow & & & \leq \\ \forall y_1 & \rightarrow & \exists y_2 & & \forall y_3 & \rightarrow & \dots & v(y_1, \dots, y_k) \end{array}$$

Players \exists and \forall play two copies of the game. In the second copy they switch the roles, i.e., \exists maximizes and \forall minimizes.

The sentence says that

\exists can achieve nonnegative pay-off: what he gains in the second game is at least what he loses in the first game.

Note that player \forall can always achieve $v(x_1, \dots, x_k) = v(y_1, \dots, y_k)$ by playing the copy-cat strategy. The order of moves is the most favorable for \forall and the least favorable for \exists .

A characterization of provably total search problems of $T_2^k (\leftrightarrow \Theta_{\Sigma_k^p})$

The **herbrandization** of a sentence in prenex form

$$\exists z_1 \forall z_2 \exists z_3 \forall z_4 \dots \phi(z_1, \dots, z_k)$$

is

$$\exists z_1 \exists z_3 \dots \phi(z_1, h_2(z_1), z_3, h_4(z_1, z_3), \dots),$$

where h_2, h_4, \dots are new function symbols.

Fact A sentence logically implies its herbrandization.

Game-theoretical interpretation of herbrandization is:

The player \forall is replaced by a strategy h_2, h_4, \dots .

We shall use the herbrandization of the sentence

$$\exists x_1 \forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \dots [v(x_1, \dots, x_k) \leq v(y_1, \dots, y_k)], \quad (1)$$

which axiomatizes T_2^k . The herbrandization is:

$$\exists x_1 \exists y_2 \exists x_3 \dots [v(x_1, h_2(x_1, y_2), x_3, \dots) \leq v(h_1(x_1), y_2, h_3(x_1, y_2, x_3), \dots)],$$

with h_1, \dots, h_k now being arbitrary polynomial time computable functions.

Definition

A k-PLS problem is given by polynomial time computable functions

$$v(x_0, x_1, \dots, x_k), h_1(x_0), h_2(x_0, x_1), \dots, h_k(x_0, x_1, \dots, x_k),$$

whose values are bounded by x_0 .

For a given input a , the task is to find numbers $b_1, c_2, b_3, c_4, \dots \leq a$ such that

$$v(a, b_1, h_2(a, b_1, c_2), b_3, \dots) \leq v(a, h_1(a, b_1), c_2, h_3(a, b_1, c_2, b_3), \dots)$$

- $k = 1$:

Given a , find $b \leq a$ such that

$$v(a, b) \leq v(a, h(a, b)).$$

This is essentially the usual PLS: v is the cost function, h is the neighborhood function.¹

- $k = 2$:

Given a , find $b, c \leq a$ such that

$$v(a, b, h_2(a, b, c)) \leq v(a, h_1(a, b), c).$$

To solve the problem put

$$b := \min_x \max_y v(a, x, y),$$

$$c := \max_y v(a, h_1(a, b), y).$$

¹The standard way of defining PLS uses the opposite inequality.

Theorem

For $k \geq 1$,

1. T_2^k proves that k -PLS problems are total.
2. Every total search problem S which is provably total in T_2^k is polynomially reducible to a k -PLS problem.

Proof-idea.

1. k -PLS problems are instances of the herbrandization of the sentence (1), which is provable in T_2^k .
2. One can show that over the weak basis theory S_2^1 all $\forall\Sigma_1^b$ sentences follow from instances of k -PLS. This is because the GI_k principle is reducible to k -PLS, provably in S_2^1 [this is due to Thapen].

Problem

Prove that in a relativized world $(k+1)$ -PLS is not reducible to k -PLS.

Why is this interesting?

- We can prove oracle separations of T_2^{k+1} from T_2^k , but only with respect to $\forall \Sigma_k^b$ sentences.
- We want to confirm the conjecture that there is no complete **TFNP** problem, which is equivalent to

Conjecture

For every r.e. consistent theory T , there exists a total polynomial search problem S which is strictly stronger than all search problems provably total in T .

For stronger theories it seems difficult to prove such relativized separations.

Thank you!