Blowing Holes in Various Aspects of Computational Problems, with Applications to Constraint Satisfaction

Who? Victor Lagerkvist

From? Theoretical Computer Science Laboratory, Linköpings Universitet, Sweden

When? September 10, 2013

Introduction

By a classical result from Ladner we know that if P $\neq$ NP then there exists problems which are in NP but are neither NP-hard nor in P.
Arora and Barak (*Computational Complexity: A Modern Approach*) write the following:

We do not know of a natural decision problem that, assuming $NP \neq P$, is proven to be in $NP \setminus P$ but not NP-complete, and there are remarkably few candidates for such languages

We provide a framework for obtaining NP-intermediate problems (under the assumption P $\neq$ NP) and apply it to problems parameterized by constraint languages.

NP-intermediate problems

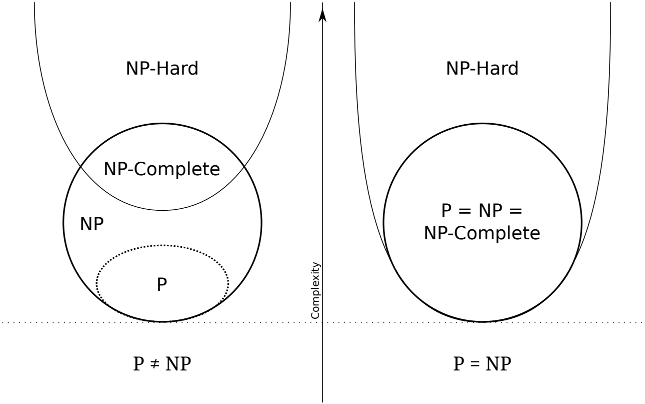

Figure : A tentative map of P versus NP.

Ladner's technique of blowing holes

Blowing holes is a diagonalization technique in which one creates a problem from an NP-complete problem X which is too sparse to be NP-hard and too dense to be in P.

- Define a *very* slowly growing function f from $\mathbb{N}$ to $\mathbb{N}$.

Ladner's technique of blowing holes

Blowing holes is a diagonalization technique in which one creates a problem from an NP-complete problem X which is too sparse to be NP-hard and too dense to be in P.

- Define a *very* slowly growing function f from $\mathbb{N}$ to $\mathbb{N}$.
- The function f is then used to blow holes in the instances of X such that the resulting problem is NP-intermediate.

Ladner's technique of blowing holes

Blowing holes is a diagonalization technique in which one creates a problem from an NP-complete problem X which is too sparse to be NP-hard and too dense to be in P.

- Define a *very* slowly growing function f from $\mathbb{N}$ to $\mathbb{N}$.
- The function f is then used to blow holes in the instances of X such that the resulting problem is NP-intermediate.
- In effect, the function f is computed by trying to find an instance which was not removed by f and, depending on the parity of the input, is either too hard to be solvable in polynomial time or to easy to make the problem NP-hard.

Preliminaries

By a *polynomial-time reduction* from problem X to problem X', we mean a Turing reduction from X to X' that runs in time $O(p(||I||))$ for some polynomial p, where $||I||$ is the number of bits required to represent the instance I in the set of instances $I(X)$.

Definition

Let X be a decision problem. A total and computable function $\rho : I(X) \to 2^{\mathbb{N}} \setminus \{\emptyset\}$ is said to be a *measure function*.
If $\rho(I)$ is a singleton set for every $I \in I(X)$, then we say that ρ is *single-valued*, and otherwise that it is *multi-valued*.

Preliminaries

Example Let X be the Boolean satisfiability problem (SAT) and let I be a formula over n variables in conjunctive normal form. Then two possible measure functions are $\rho(I) = \{n\}$ and $\sigma(I) = \{k \mid (l_1 \vee \ldots l_k) \text{ is a clause in } I\}$. Note that ρ is single-valued while σ is multi-valued.

Preliminaries

A measure function ρ together with a problem X then yields a problem $X_\rho(\cdot)$ parameterized by a set of integers S:

Instance. Instance I of X such that $\rho(I) \subseteq S$.
Question. Is I a yes-instance?

Example

Let E be the set of even numbers. With the measure functions ρ and σ from the previous example we see that $\text{SAT}_\rho(E)$ is the SAT problem restricted to instances with an even number of variables, while $\text{SAT}_\sigma(E)$ is the SAT problem restricted to instances with even clause lengths.

Our framework

Theorem

Let $X_\rho(\cdot)$ be a computational decision problem with a measure function ρ. Assume that $X_\rho(\cdot)$ and $S \subseteq \mathbb{N}$ satisfies the following properties:

P0: $I(X)$ is recursively enumerable.
P1: $X_\rho(S)$ is NP-complete.
P2: $X_\rho(T)$ is in P whenever T is a finite subset of S.
P3: $X_\rho(S)$ is polynomial-time reducible to $X_\rho(T)$ whenever $T \subseteq S$ and $S \setminus T$ is finite.

Then, if $\mathrm{P} \neq \mathrm{NP}$, there exists a set $S' \subset S$ such that $X_\rho(S')$ is in $\mathrm{NP} \setminus \mathrm{P}$ and $X_\rho(S)$ is not polynomial-time reducible to $X_\rho(S')$.

Example: Ladner's result

Assume $\mathrm{P} \neq \mathrm{NP}$. Let X be any r.e. NP-complete problem, e.g. SAT. Then for every instance I define the measure function ρ such that $\rho(I) = ||I||$ and let $S = \mathbb{N}$. Note that ρ is **single-valued**.

Example: Ladner's result

Assume $\mathrm{P} \neq \mathrm{NP}$. Let X be any r.e. NP-complete problem, e.g. SAT. Then for every instance I define the measure function ρ such that $\rho(I) = ||I||$ and let $S = \mathbb{N}$. Note that ρ is **single-valued**.

- $X_\rho(\mathbb{N})$ is then the same problem as X. This immediately proves that $X_\rho(\mathbb{N})$ is NP-complete. (Property P1.)

Example: Ladner's result

Assume $\mathrm{P} \neq \mathrm{NP}$. Let X be any r.e. NP-complete problem, e.g. SAT. Then for every instance I define the measure function ρ such that $\rho(I) = ||I||$ and let $S = \mathbb{N}$. Note that ρ is **single-valued**.

- $X_\rho(\mathbb{N})$ is then the same problem as X. This immediately proves that $X_\rho(\mathbb{N})$ is NP-complete. (Property P1.)
- Let U be a finite subset of $\mathbb{N}$. Then $X_\rho(U)$ only has a finite set of instances and can therefore be solved in constant time. (Property P2.)

Example: Ladner's result

Assume $\mathrm{P} \neq \mathrm{NP}$. Let X be any r.e. NP-complete problem, e.g. SAT. Then for every instance I define the measure function ρ such that $\rho(I) = ||I||$ and let $S = \mathbb{N}$. Note that ρ is **single-valued**.

- $X_\rho(\mathbb{N})$ is then the same problem as X. This immediately proves that $X_\rho(\mathbb{N})$ is NP-complete. (Property P1.)
- Let U be a finite subset of $\mathbb{N}$. Then $X_\rho(U)$ only has a finite set of instances and can therefore be solved in constant time. (Property P2.)
- Let $U \subset \mathbb{N}$ such that $\mathbb{N} \setminus U = \{u_1, \ldots, u_k\}$ is finite. Since every $X_\rho(u_i)$ is solvable in constant time it is easy to reduce $X_\rho(\mathbb{N})$ to $X_\rho(U)$. (Property P3).

Example: Ladner's result

Assume $\mathrm{P} \neq \mathrm{NP}$. Let X be any r.e. NP-complete problem, e.g. SAT. Then for every instance I define the measure function ρ such that $\rho(I) = ||I||$ and let $S = \mathbb{N}$. Note that ρ is **single-valued**.

- $X_\rho(\mathbb{N})$ is then the same problem as X. This immediately proves that $X_\rho(\mathbb{N})$ is NP-complete. (Property P1.)
- Let U be a finite subset of $\mathbb{N}$. Then $X_\rho(U)$ only has a finite set of instances and can therefore be solved in constant time. (Property P2.)
- Let $U \subset \mathbb{N}$ such that $\mathbb{N} \setminus U = \{u_1, \ldots, u_k\}$ is finite. Since every $X_\rho(u_i)$ is solvable in constant time it is easy to reduce $X_\rho(\mathbb{N})$ to $X_\rho(U)$. (Property P3).

By the main theorem it then follows that there exists a $T \subset \mathbb{N}$ such that $X_\rho(T)$ is NP-intermediate.

Example: Subset-sum

The Subset-Sum problem is one of Karp's classical problems and can be defined as follows.

Instance: A finite set $Y \subseteq \mathbb{N}$ and a number $k \in \mathbb{N}$.
Question: Is there a $Y' \subseteq Y$ such that $\sum Y' = k$?

Example: Subset-sum

The Subset-Sum problem is one of Karp's classical problems and can be defined as follows.

Instance: A finite set $Y \subseteq \mathbb{N}$ and a number $k \in \mathbb{N}$.
Question: Is there a $Y' \subseteq Y$ such that $\sum Y' = k$?

We define a **multi-valued** measure function ρ such that $\rho((Y,k)) = Y$ and let $S = \mathbb{N}$. Property P0 and P1 then follows analogously to the previous example. For property P2 simply note that Subset-Sum is solvable in pseudopolynomial time with respect to the difference between the largest and smallest number in Y. Property P3 is trickier and involves a Turing reduction which makes sure that the removed elements does not affect the existence of a solution.

Blowing holes depends on the parameter

Our observation is that the properties of X_ρ depends heavily on whether ρ is single- or multi-valued. Property P0 and P1 are trivial for most problems but the remaining properties depend heavily on the parameter in question.

Blowing holes depends on the parameter

Our observation is that the properties of X_ρ depends heavily on whether ρ is single- or multi-valued. Property P0 and P1 are trivial for most problems but the remaining properties depend heavily on the parameter in question.

- For single-valued measure functions property P2 is essentially the same as finding a parameter such that the problem is fixed-paramater tractable (in FPT). This also implies that property P3 is satisfied.

Blowing holes depends on the parameter

Our observation is that the properties of X_ρ depends heavily on whether ρ is single- or multi-valued. Property P0 and P1 are trivial for most problems but the remaining properties depend heavily on the parameter in question.

- For single-valued measure functions property P2 is essentially the same as finding a parameter such that the problem is fixed-paramater tractable (in FPT). This also implies that property P3 is satisfied.
- However, the real difficulty often lies in defining ρ in such a way that $X_\rho(\cdot)$ becomes an interesting problem. We claim that multi-valued measure functions often yields better results and demonstrate it on *constraint satisfaction problems* and the *propositional abduction problem*.

Constraint satisfaction problems

The *constraint satisfaction problem* over Γ (abbreviated as CSP(Γ)) is defined as follows.

Instance: A set V of variables and a set C of constraint applications $R(v_1, \ldots, v_k)$ where $R \in \Gamma$, $k = ar(R)$, and $v_1, \ldots, v_k \in V$.
Question: Is there a total function $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for each constraint $R(v_1, \ldots, v_k)$ in C?

Constraint satisfaction problems over finite domains

The non-existence of NP-intermediate problems for the CSP problem over finite domains is known as the *Feder-Vardi* conjecture and is known to hold for domain size 2 (Schaefer), 3 (Bulatov) and 4(Markovic et al, not yet publicised).
The conjecture, if true, would give a very large and natural subclass of NP which does not admit any NP-intermediate problems.

Constraint satisfaction problems over infinite domains

- For infinite domains the situation differs: Bodirsky and Grohe proved that *all* computational decision problems are reducible to a constraint satisfaction problem over an infinite domain.
- This trivially show that the full CSP(Γ) problem does not have a dichotomy between P and NP-complete. However, the language constructed in such a reduction is arguably even harder to make sense of than the original NP-intermediate problem.

Constraint satisfaction problems over infinite domains

We give a constraint language Γ such that:

- CSP(Γ) is NP-intermediate.
- Γ contains only ternary relations closely related to integer programming problems over the natural numbers.
- CSP(Δ) is tractable for every finite $\Delta \subset \Gamma$.

In addition we show that all constraint languages Γ, over finite or infinite domains, can be extended in such a way that the reducibility property P3 is automatically fulfilled. For finite domains this requires an infinite structure but for infinite domains the arity of relations is only increased by 1.

The propositional abduction problem

- Let Γ denote a constraint language. The propositional abduction problem $\mathrm{Abd}(\Gamma)$ is a fundamental form of non-monotonic reasoning in which we are given a manifestation and try to find an explanation.

The propositional abduction problem

- Let Γ denote a constraint language. The propositional abduction problem $\mathrm{Abd}(\Gamma)$ is a fundamental form of non-monotonic reasoning in which we are given a manifestation and try to find an explanation.
- Well studied from a complexity theoretical point of view. Nordh and Zanuttini proved a classification for *finite* constraint languages Γ and proved that $\mathrm{Abd}(\Gamma)$ is either in P, NP-complete, Co-NP-complete or Σ_2^P-complete.

The propositional abduction problem

- Let Γ denote a constraint language. The propositional abduction problem $\mathrm{Abd}(\Gamma)$ is a fundamental form of non-monotonic reasoning in which we are given a manifestation and try to find an explanation.
- Well studied from a complexity theoretical point of view. Nordh and Zanuttini proved a classification for *finite* constraint languages Γ and proved that $\mathrm{Abd}(\Gamma)$ is either in P, NP-complete, Co-NP-complete or Σ_2^P-complete.
- For *infinite* constraint languages the situation differ. We give an infinite constraint language Γ such that $\mathrm{Abd}(\Gamma)$ is NP-intermediate.

Research directions and open questions

- Issue a more systematic study of multi-valued measure functions.
- In particular problems parameterized by constraint languages since they are closely related to CSPs.
- Can a similar framework based on other complexity-theoretical assumptions, e.g. the exponential-time hypothesis, be meaningful?