Theoretical Computer Science 581 (2015) 67–82

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Constructing NP-intermediate problems by blowing holes with parameters of various properties ☆

Peter Jonsson [a], Victor Lagerkvist [b,*], Gustav Nordh [b]

[a] Department of Computer and Information Science, Linköpings University, Sweden
[b] Kvarnvägen 6, 53374, Hällekis, Sweden

ARTICLE INFO

Article history:
Received 2 June 2014
Received in revised form 10 February 2015
Accepted 3 March 2015
Available online 6 March 2015
Communicated by M. Kiwi

Keywords:
Computational complexity
NP-intermediate problems
Constraint satisfaction problems
Abduction problems

ABSTRACT

The search for natural NP-intermediate problems is one of the holy grails within computational complexity. Ladner's original diagonalization technique for generating NP-intermediate problems, blowing holes, has a serious shortcoming: it creates problems with a highly artificial structure by arbitrarily removing certain problem instances. In this article we limit this problem by generalizing Ladner's method to use parameters with various characteristics. This allows one to define more fine-grained parameters, resulting in NP-intermediate problems where we only blow holes in a controlled subset of the problem. We begin by fully characterizing the problems that admit NP-intermediate subproblems for a broad and natural class of parameterizations, and extend the result further such that structural CSP restrictions based on parameters that are hard to compute (such as tree-width) are covered, thereby generalizing a result by Grohe. For studying certain classes of problems, including CSPs parameterized by constraint languages, we consider more powerful parameterizations. First, we identify a new method for obtaining constraint languages Γ such that CSP(Γ) are NP-intermediate. The sets Γ can have very different properties compared to previous constructions (by, for instance, Bodirsky & Grohe) and provides insights into the algebraic approach for studying the complexity of infinite-domain CSPs. Second, we prove that the propositional abduction problem parameterized by constraint languages admits NP-intermediate problems. This settles an open question posed by Nordh & Zanuttini.

1. Introduction

1.1. Background

Assuming P $\neq$ NP it is natural to consider problems in NP $\setminus$ P which are not NP-hard. Such problems are referred to as *NP-intermediate*, and Ladner [30] explicitly constructed NP-intermediate problems by removing strings of certain lengths from NP-complete languages via a diagonalization technique that is colloquially known as *blowing holes in problems*. The languages constructed via blowing are unfortunately famous for being highly artificial: Arora and Barak [3] write the following.

☆ A preliminary version of this article appeared in *Proceedings of the 19th International Conference on Principles and Practice of Constraint Programming* (CP-2013), pp. 398–414. Uppsala, Sweden, Sep., 2013.

* Corresponding author.
E-mail addresses: peter.jonsson@liu.se (P. Jonsson), victor.lagerkvist@liu.se (V. Lagerkvist), gustav.nordh@gmail.com (G. Nordh).

http://dx.doi.org/10.1016/j.tcs.2015.03.009
0304-3975/

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.

More natural examples are known under other complexity-theoretic assumptions. For instance, LogClique (the problem of deciding whether an n-vertex graph contains a clique of size $\log n$) is NP-intermediate under the exponential-time hypothesis (ETH). We wish to stress the difference between problems that, assuming P $\neq$ NP, are provably neither P nor NP-complete, and problems whose complexity is simply undetermined at the moment. As for the latter, of the dozen problems in Garey & Johnson [20] which at the time where not known to be P or NP-complete, only a few, such as the integer factorization problem and the graph isomorphism problem, remain unresolved. The integer factorization problem is particularly interesting in this sense: it is not likely to be NP-complete since it is both in NP and coNP, and no polynomial-time algorithm is known despite considerable efforts to construct one. The lack of natural NP-intermediate computational problems makes it important to investigate new classes of NP-intermediate problems and, hopefully, increase our understanding of the borderline between P and NP.

In the "opposite direction", there have been attempts to isolate subclasses of NP which exhibit dichotomies between P and NP-complete, i.e. non-trivial subclasses that do not admit NP-intermediate problems. For instance, Feder and Vardi [18] conjectured that the *constraint satisfaction problem* (CSP) over finite domains exhibits such a dichotomy, but so far the conjecture is only known to hold for domains of two and three elements, as proven by Schaefer [38] and Bulatov [7], respectively. One of the reasons behind this conjecture is that the constraint satisfaction problem is included in *monotone monadic SNP without inequality* (MMSNP), where SNP is a subset of NP characterizable through a special class of existential second-order sentences, and it is known that adding only marginally more expressive sentences to MMSNP results in a non-dichotomizable complexity class [18]. The set MMSNP is therefore viewed as a candidate as a maximal subclass of NP which does not contain any problems of intermediate complexity. From this it clearly follows that some restrictions, e.g. constraint language restrictions, are regarded as more interesting than removing arbitrary strings from the set of valid instances as in Ladner's original proof. In other words the existence of NP-intermediate subproblems depends heavily on which *parameter* one chooses to restrict and it is safe to say that we currently lack a deeper understanding of which parameters one can use to find NP-intermediate subproblems, and when dichotomies arise. Hence, we propose a strategy consisting of investigating subclasses of NP induced by different parameters, in order to determine which problems admit dichotomies and which do not, and increase our understanding of the puzzling nature of intermediate problems.

1.2. Article structure

We begin (in Section 3) by presenting a diagonalization method for obtaining NP-intermediate problems, based on parameterizing decision problems in different ways. In our framework, a parameter, or a *measure function*, is simply a computable function ρ from the instances of some decision problem X to the non-empty subsets of $\mathbb{N}$. We say that such a function is *single-valued* if $\rho(I)$ is a singleton set for every instance of X, and *multi-valued* otherwise. Depending on the parameter one obtains problems with different characteristics. Simple applications of our method include the connection between the complexity class XP and NP-intermediate problems observed by Chen et al. [12]. Even though our method is still based on diagonalization we claim that the intermediate problems obtained are qualitatively different from the ones obtained by Ladner's original method, and that they can be used for gaining new insights into the complexity of computational problems. Whether a problem is "natural" or not is of course highly subjective and a matter of taste, but there is a wider consensus that some types of restrictions, such as constraint language restrictions, are more interesting than others. Throughout this article we will see that our diagonalization framework in combination with different measure functions allows us to construct NP-intermediate problems also for such non-trivial cases, which may constitute new and interesting sources of intermediate problems.

In Section 4, we analyze the applicability of the diagonalization method for single-valued measure functions. Under mild additional assumptions, we obtain a full understanding of when NP-intermediate problems arise when the measure function is single-valued and polynomial-time computable. We also relate the structure of subproblems induced by single-valued measure functions to the question of whether the set of all NP-intermediate problems is closed under disjoint union. Unfortunately, CSPs under structural restrictions (i.e. when considering instances with bounded width parameters) are not captured by these results since width parameters are typically not polynomial-time computable. To remedy this, we present a general method for obtaining NP-intermediate problems based on structurally restricted CSPs in Section 4.3. This is a generalization of a result by Grohe [22] who has shown that, under the assumption that FPT $\neq$ W[1], NP-intermediate CSP problems can be obtained by restricting the tree-width of their corresponding primal graphs. Our result implies that this holds also under the weaker assumption that P $\neq$ NP and for many width parameters. NP-intermediate problems based on structural restrictions have also been identified by Bodirsky & Grohe [5].

Multi-valued measure functions are apparently harder to study and a full understanding appears difficult to obtain. We first relate single-valued measure functions with multi-valued measure functions (in Section 5.1) and show that every multi-valued measure function effectively determines a single-valued measure function which shares many fundamental properties. Despite this, single-valued measure functions have limited applicability when studying problems parameterized by constraint languages, such as constraint satisfaction problems, and we give several examples which highlight why this is the case. For problems parameterized by constraint languages we therefore focus exclusively on multi-valued measure