

# Fine-Grained Algorithms and Complexity

Virginia Vassilevska Williams

Stanford University, USA  
virgi@cs.stanford.edu

---

## Abstract

A central goal of algorithmic research is to determine how fast computational problems can be solved in the worst case. Theorems from complexity theory state that there are problems that, on inputs of size  $n$ , can be solved in  $t(n)$  time but not in  $t(n)^{1-\epsilon}$  time for  $\epsilon > 0$ . The main challenge is to determine where in this hierarchy various natural and important problems lie. Throughout the years, many ingenious algorithmic techniques have been developed and applied to obtain blazingly fast algorithms for many problems. Nevertheless, for many other central problems, the best known running times are essentially those of the classical algorithms devised for them in the 1950s and 1960s.

Unconditional lower bounds seem very difficult to obtain, and so practically all known time lower bounds are conditional. For years, the main tool for proving hardness of computational problems have been NP-hardness reductions, basing hardness on  $P \neq NP$ . However, when we care about the exact running time (as opposed to merely polynomial vs non-polynomial), NP-hardness is not applicable, especially if the problem can already be solved in polynomial time. In recent years, a new theory has been developed, based on “fine-grained reductions” that focus on exact running times. The goal of these reductions is as follows. Suppose problem A is solvable in  $a(n)$  time and problem B in  $b(n)$  time, and no  $a(n)^{1-\epsilon}$  and  $b(n)^{1-\epsilon}$  algorithms are known for A and B respectively, for any  $\epsilon > 0$ . Then if A is fine-grained reducible to problem B (for  $a(n)$  and  $b(n)$ ), then a  $b(n)^{1-\epsilon}$  time algorithm for B (for any  $\epsilon > 0$ ) implies an  $a(n)^{1-\epsilon'}$  algorithm for A (for some  $\epsilon' > 0$ ). Now, mimicking NP-hardness, the approach is to (1) select a key problem X that is conjectured to require  $t(n)^{1-o(1)}$  time for some  $t(n)$ , and (2) reduce X in a fine-grained way to many important problems. This approach has led to the discovery of many meaningful relationships between problems, and even sometimes to equivalence classes.

In this talk I will give an overview of the current progress in this area of study, and will highlight some new exciting developments.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems, F.1.3 Complexity Measures and Classes

**Keywords and phrases** fine-grained reductions, polynomial time problems, equivalence, hardness

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2016.3

**Category** Invited Talk



© Virginia V. Williams;  
licensed under Creative Commons License CC-BY  
33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016).  
Editors: Nicolas Ollinger and Heribert Vollmer; Article No. 3; pp. 3:1–3:1  
Leibniz International Proceedings in Informatics  
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

