Exact Complexity and Satisfiability
(Invited Talk)

Russell Impagliazzo[⋆] and Ramamohan Paturi[⋆]

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404, USA
{russell,paturi}@cs.ucsd.edu

All **NP**-complete problems are equivalent as far as polynomial time solvability is concerned. However, their exact complexities (worst-case complexity of algorithms that solve every instance correctly and exactly) differ widely. Starting with Bellman [1], Tarjan and Trojanowski [9], Karp [5], and Monien and Speckenmeyer [7], the design of improved exponential time algorithms for **NP**-complete problems has been a tremendously fruitful endeavor, and one that has recently been accelerating both in terms of the number of results and the increasing sophistication of algorithmic techniques. There are a vast variety of problems where progress has been made, e.g., k-SAT, k-COLORABILITY, MAXIMUM INDEPENDENT SET, HAMILTONIAN PATH, CHROMATIC NUMBER, and CIRCUIT SAT for limited classes of circuits. The "current champion" algorithms for these problems deploy a vast variety of algorithmic techniques, e.g., back-tracking search (with its many refinements and novel analysis techniques), divide-and-conquer, dynamic programming, randomized search, algebraic transforms, inclusion-exclusion, color coding, split and list, and algebraic sieving. In many ways, this is analogous to the diversity of approximation ratios and approximation algorithms known for different **NP**-complete problems. In view of this diversity, it is tempting to focus on the distinctions between problems rather than the interconnections between them. However, over the past two decades, there has been a wave of research showing that such connections do exist. Furthermore, progress on the exact complexity of **NP**-complete problems is linked to other fundamental questions in computational complexity, such as circuit lower bounds, parameterized complexity, data structures, and the precise complexity of problems within **P**. We are honored that our work helped to catalyze this wave of research, and humbled by the extent to which later researchers went far beyond our dreams of what might be possible.

We are severely constrained by space restrictions here, so we will not be able to give due credit or even describe many of the significant results. We will confine ourselves to informally defining the questions to be investigated, describing the initial steps to answer them, and giving some open problems.

The basic issue is to understand the extent to which "exhaustive search" can be beaten. However, for a decision problem, there might be several ways to

[⋆] This research is supported by NSF grant CCF-1213151 from the Division of Computing and Communication Foundations.

associate a search problem, and hence a trivial algorithm. We decided to finesse this issue by utilizing parameterized complexity [2], introducing a *complexity parameter* in addition to the usual size parameter. If the number of bits required to describe solutions is bounded by the complexity parameter, then we use the label "brute-force" to describe an algorithm that is exponential in the complexity parameter and polynomial in the size. For example, for the k-SAT problem, we will typically use the number of variables as a complexity parameter. However, we could also consider another parameterization for the k-SAT problem where the number of clauses is the complexity parameter. In either case, the size of the instance, in terms of bits used to represent it, will typically be significantly larger than this parameter.

For **NP**-hard problems with input size m and complexity parameter n, we call an algorithm *brute-force* if it runs in time $\texttt{poly}(m)2^n$. We say that algorithm is an *improvement* if its running time is bounded by $\texttt{poly}(m)2^{\mu n}$ for some $\mu < 1$. An *exponential* improvement is one where $\mu \in 1 - \Omega(1)$, and a *nontrivial* improvement is one where $\mu = 1 - \omega(\log m/n)$. If $\mu = o(1)$, then the algorithm is *sub-exponential.*

Many **NP**-complete problems have exponential improvements, and a few such as chromatic number or Hamiltonian path for planar graphs have subexponential improvements. On the other hand, for problems like CIRCUIT SAT, we do not even know of any nontrivial improvements. It is not clear whether the series of improvements for problems such as k-SAT and k-COLORABILITY will lead to subexponential algorithms. For what problems, can we expect improved algorithms? Is progress on various problems connected? Can we give complexity-theoretic reasons why improvements or further improvements might not be possible for some problems?

To understand the difficulty of connecting the exact complexities, consider the standard reduction from k-SAT to k-COLORABILITY. It maps a k-SAT instance F of size m and n variables to a graph on $O(m+n)$ variables and $O(m+n)$ vertices. Since $m = O(n^k)$, a subexponential time algorithm for k-COLORABILITY would not a priori let us conclude anything useful about k-SAT.

In [4], progress was made to resolve this issue where it is shown subexponentital time solvability of both the problems is equivalent. The key new ideas include the notion of subexponential time Turing reductions and the Sparsification Lemma, which states that any k-CNF can be expressed as the subexponential disjunction of linear (in the number of variables) size k-CNF in subexponential time. From this lemma, we can derive a way to convert a subexponential time algorithm for k-COLORABILITY into a subexponential time algorithm for k-SAT. More generally, in [4], it is shown that all the **SNP**-complete problems (under subexponential time Turing reductions), are equivalent as far as subexponential time complexities are concerned. **SNP** is a subclass of **NP** which includes k-SAT and k-COLORABILITY, and is defined as the class of problems expressible by second order existential quantifiers followed by a first order universal quantifiers followed by a basic formula, introduced by Papadimitriou and Yannakakis [8]. The result also extends to size-constrained **SNP** where the second order quantified variables are restricted in size.

The result provides evidence that k-SAT (equivalently, 3-SAT) may not have subexponential time algorithms since otherwise the entire *logically* defined class **SNP** would have subexponential time algorithms. While we are not in a position to resolve this issue, it is interesting to explore the state of affairs assuming the likelihood. Let $s_k = \inf\{\delta | \exists\ 2^{\delta n}$ algorithm for k-SAT$\}$. We define the *Exponential Time Hypothesis* (**ETH**) to be the statement: $s_3 > 0$. We call it a hypothesis rather than a conjecture, because we are agnostic about whether it is actually true, but think that its truth value has interesting ramifications either way.

We understand very little about exponential time algorithms. **ETH** will be useful if it helps factor out the essential difficulty of dealing with exponential time algorithms for **NP**-complete problems. More concretely, the usefulness of **ETH** is its explanatory value regarding the exact complexities of various **NP**-complete problems, ideally, by providing lower bounds that match the best known upper bounds.

One of the first nontrivial consequences of **ETH** is that the exponential complexities s_k of k-SAT strictly increase infinitely often as k increases [3]. This result tempts one to posit the *Strong Exponential Time Hypothesis* (**SETH**) that says $s_\infty := \lim_{k\to\infty} s_k = 1$. There have been numerous lower bounds on the exact complexities of **NP**-complete problems based on **ETH** and **SETH**. Lokshtanov, Marx, and Saurabh provide a systematic summary of these in results in [6].

However, a number of questions remain regarding the explanatory power of **ETH**. Assuming **ETH** or other well known complexity assumption, can we obtain a positive constant lower bound on s_3? Can we prove that **ETH** implies **SETH**? Assuming **SETH**, can we prove a $2^{(n-o(n))}$ lower bound for CHROMATIC NUMBER?

References

1. Bellman, R.: Dynamic programming treatment of the travelling salesman problem. Journal of the ACM 9(1), 61–63 (1962)
2. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1999)
3. Impagliazzo, R., Paturi, R.: The complexity of k-SAT. Journal of Computer and Systems Sciences 62(2), 367–375 (2001); Preliminary version in 14th Annual IEEE Conference on Computational Complexity, pp. 237–240 (1999)
4. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? Journal of Computer and System Sciences 63, 512–530 (1998); Preliminary version. In: 39th Annual IEEE Symposium on Foundations of Computer Science, pp. 653–662 (1998)
5. Karp, R.M.: Dynamic programming meets the principle of inclusion and exclusion. Operations Research Letters 1, 49–51 (1982)
6. Lokshtanov, D., Marx, D., Saurabh, S.: Lower bounds based on the exponential time hypothesis. Bulletin of the EATCS 105, 41–72 (2011)
7. Monien, B., Speckenmeyer, E.: Solving satisfiability in less than 2^n steps. Discrete Applied Mathematics 10, 287–295 (1985)
8. Papadimitriou, C., Yannakakis, M.: Optimization, approximation, and complexity classes. Journal of Computer and System Sciences 43, 425–440 (1991)
9. Tarjan, R., Trojanowski, A.: Finding a maximum independent set. SIAM Journal of Computing 6, 537–546 (1977)