

Speedup for Natural Problems and Noncomputability[☆]

Hunter Monroe

International Monetary Fund, 700 19th St, NW, Washington, DC 20431

Abstract

A resource-bounded version of the statement “no algorithm recognizes all non-halting Turing machines” is equivalent to an infinitely often (i.o.) super-polynomial speedup for the time required to accept any (paddable) **coNP**-complete language and also equivalent to a superpolynomial speedup in proof length in propositional proof systems for tautologies, each of which implies $\mathbf{P} \neq \mathbf{NP}$. This suggests a correspondence between the properties “has no algorithm at all” and “has no best algorithm” which seems relevant to open problems in computational and proof complexity.

Key words: Speedability, Speedup

1. Introduction

Informally, a language L has speedup if, for any Turing machine (TM) for L , there exists one that is better. Blum [2] exhibited languages that have almost-everywhere speedup, which are unnatural being constructed solely for that purpose. The possibility of weaker speedups for natural languages has received less attention [15]. Some suspect that integer multiplication and matrix multiplication (MM) have a slight, superlinear speedup [18, 14, 3],

[☆]This paper is in honor of the retirement of Benjamin Klein from Davidson College. The views expressed in this column are those of the author and should not be attributed to the International Monetary Fund, its Executive Board, or its management. This paper could not have been prepared without encouragement from Marius Zimand and Bill Gasarch. This paper arose from a suggestion by Richard Beigel. I would also like to thank an anonymous referee, Scott Aaronson, Amir Ben-Amram, Neil Christensen, Lance Fortnow, Jörg Flum, Yijia Chen, and participants in a University of Maryland Complexity Seminar who provided useful comments. Remaining errors are my own.

Email address: `hmonroe.tcs@huntermonroe.com` (Hunter Monroe)

reflecting in part the large number of algorithms for these problems—about 13 and 18 respectively [1, 17]. In fact, there is no best Strassen-style bilinear MM identity [7].

We identify an intuitive condition which, like several others in the literature, is equivalent to an infinitely often (i.o.) superpolynomial speedup for the time required to accept any (paddable) **coNP**-complete language and also equivalent to a superpolynomial speedup in proof length in propositional proof systems for tautologies, each of which implies $\mathbf{P} \neq \mathbf{NP}$. This condition is a resource-bounded version of the statement “no algorithm recognizes all non-halting TMs”, suggesting a correspondence between the properties “has no algorithm at all” and “has no best algorithm” which seems relevant to open problems in computational and proof complexity.

2. Speedup for **coNP**-Complete Languages

Consider this well-known fact from computability theory:

Fact 2.1. *Given any TM M accepting only “non-halting” $\langle N, x \rangle$ for which TM N does not halt on input x , M fails to accept some particular non-halting $\langle N', x' \rangle$. In other words, the set of non-halting $\langle N, x \rangle$ is not computably enumerable (c.e.).*

By implication, there is a better TM M' that correctly accepts more non-halting inputs than M by accepting the input $\langle N', x' \rangle$ and otherwise running M . This section considers a corresponding resource-bounded statement in complexity theory regarding N which do not halt on x within t steps.

Notation: M and M' will denote deterministic TMs throughout the paper, and, henceforth, N and N' will denote nondeterministic TMs. Define $\mathbf{BHP} = \{\langle N, x, 1^t \rangle \mid \text{there is at least one accepting path of nondeterministic TM } N \text{ on input } x \text{ with } t \text{ or fewer steps}\}$ and define $\mathbf{coBHP} = \{\langle N, x, 1^t \rangle \mid \langle N, x, 1^t \rangle \notin \mathbf{BHP}\}$. If M is a deterministic TM, then T_M is the function that maps a string x to how many steps $M(x)$ takes. Say that M accepts a language L if M halts in an accepting state if and only if $x \in L$; M may not halt on $x \notin L$. Say that $\langle N', x' \rangle$ is non-halting if N' has no accepting path on input x' , in which case Fact 2.1 continues to hold for nondeterministic N . Note that \mathbf{BHP} is **NP**-complete with the accepting path of N on x as a certificate, and that \mathbf{coBHP} is **coNP**-complete.

The following condition corresponds to Fact 2.1:

(*) For any M accepting coBHP , there exists some non-halting $\langle N', x' \rangle$ such that the function $f(t) = T_M(N', x', 1^t)$ is not bounded by any polynomial.¹

Supposing $\mathbf{P} \neq \mathbf{NP}$ and therefore $\text{coBHP} \notin \mathbf{P}$, condition (*) rules out the absurd possibility that some M nevertheless can accept the subset of inputs beginning with any particular machine-input pair within a polynomial bound (for that subset). An intuition for why this condition might hold could be a belief that there is at least one $\langle N', x' \rangle$ for which M must infinitely often use brute force to rule out all possible accepting paths of N' on x' with at most t steps.² Under (*), coBHP has an i.o. superpolynomial speedup, defined as follows:³

Def 2.2. For M and M' accepting a language L , write $M' \leq_p M$ if there exists a polynomial p such that for all inputs $x \in L$,

$$T_{M'}(x) \leq p(|x|, T_M(x)). \quad (1)$$

If $M' \leq_p M$ but it is not the case that $M \leq_p M'$, write $M' <_p M$. If L has a least element M under $<_p$, say that M is p -optimal [12] and otherwise that L has (i.o.) superpolynomial speedup.

It is shown below that (*) is equivalent to a superpolynomial speedup for coBHP . This conclusion is significant, as superpolynomial speedup for accepting a particular (paddable) coNP -complete language is in fact equivalent to superpolynomial speedup for accepting any (paddable) coNP -complete language.⁴ Furthermore, it is also equivalent to a superpolynomial speedup for

¹The function f may depend on M , N' , and x' . For inputs not in coBHP , M does not accept, but otherwise its behavior is not constrained.

²Condition (*) is equivalent to the statement that there is no M deciding BHP within time $O(t^{f(|N', x'|)})$. Chen and Flum [4] show that under certain complexity theoretic assumptions, there is no such M for f computable.

³By contrast, Hirsch and Itsykson [9] exhibit a p -optimal heuristic randomized algorithm for accepting the set of tautologies (TAUT), where the algorithm is allowed to accept non-tautologies erroneously with bounded probability. Levin [13] exhibits a p -optimal witness search algorithm for any language in \mathbf{NP} . Levin's algorithm dovetails every possible TM, runs any output produced through a predetermined witness verifier, and then prints out the first witness that is verified. However, even though $\text{SAT} \in \mathbf{NP}$, Köbler and Messner [11] argue that accepting SAT is likely to have superpolynomial speedup.

⁴All known coNP -complete languages are paddable.

proof length in propositional proof systems for the set of tautologies (**TAUT**) [12], defined as follows. A propositional proof system is a function $h \in \mathbf{FP}$ with range **TAUT** [6]. The proof system h is p -optimal if for any other proof system f , there exists $g \in \mathbf{FP}$ such that $h(g(x)) = f(x)$ [12]. Thus, (*) holds iff there is no p -optimal propositional proof system, so propositional proof systems have a superpolynomial speedup for proof length.

Theorem 2.3. *The condition (*) holds if and only if **coBHP** has superpolynomial speedup.*

Proof: \Rightarrow Suppose condition (*) holds. Given M accepting **coBHP**, choose N', x' for M in (*), so $f(t) = T_M(\langle N', x', 1^t \rangle)$ is not polynomially bounded. We create M' as follows:

1. Input $\langle N, x, 1^t \rangle$.
2. If $\langle N, x \rangle \neq \langle N', x' \rangle$ then run $M(\langle N, x, 1^t \rangle)$.
3. If $\langle N, x \rangle = \langle N', x' \rangle$ then accept immediately.

Then $M' <_p M$, so **coBHP** has superpolynomial speedup.

\Leftarrow The converse follows from results of Chen and Flum;⁵ an anonymous referee proposed the following more direct argument. If (*) does not hold, it will be shown that there is a TM M_{opt} which is p -optimal for **coBHP**. The strategy employed by this machine is to enumerate and simulate a limited number of TMs M_i on the input of M_{opt} and accept if any M_i accepts, after verifying that this M_i accepts correctly. Crucial to the strategy are: (1) the existence of a nondeterministic machine N_c which is used to verify that M_i accepts correctly, and (2) the existence if (*) fails of an M^* which can efficiently simulate N_c .

Assume that (*) does not hold. That is, there exists some deterministic machine M^* for **coBHP** such that for any non-halting $\langle N', x' \rangle$, there exists some polynomial $p_{N', x'}$ such that M^* accepts $\langle N', x', 1^t \rangle$ in at most $p_{N', x'}(t)$ steps for all t . Assume some enumeration M_1, M_2, \dots of deterministic machines and assume that the first machine M_1 in the enumeration is a standard 2^{cn} -time machine accepting **coBHP** by brute force. Consider the following deterministic machine M_{opt} :

⁵If (*) does not hold, then **coBHP** $\in \mathbf{XP}_{uni}$, where a parameterized problem (Q, κ) is in \mathbf{XP}_{uni} if there is an M deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ (for **coBHP**, $\langle N, x \rangle$ is the parameter κ). In that case, there is a p -optimal M accepting any **coNP**-complete language, including **coBHP** ([5] Theorem 8 and Lemma 18).

1. Input $y = \langle N, x, 1^t \rangle$ for the **coBHP** problem (let $n = |y|$);
2. For each $\tau = n, n + 1, \dots$, run all machines M_1, \dots, M_n on y within τ steps:
 - If M_1 terminates and accepts y , then accept y and halt;
 - If some M_i accepts y , then accept y and halt after verifying that:
 - (**) There is no instance of **BHP** of length $\leq \tau$ such that M_i wrongly accepts it in τ steps.

Clearly M_{opt} accepts **coBHP** and otherwise does not halt.

The key idea is to reduce the problem of checking (**) to some halting problem in **coBHP**. Consider the execution of the following nondeterministic machine N_c :

1. Input $x' = M_i$;
2. For each $\tau' = 1, 2, \dots$,
 - Guess $z (= \langle N, x, 1^t \rangle)$ of length $\leq \tau'$ and w in $\{0, 1\}^{\tau'}$;
 - If M_i accepts z within τ' steps and w witnesses z in **BHP**, then accept x' and halt.

Then $\langle N_c, M_i \rangle$ is non-halting if M_i accepts **coBHP**. Furthermore, there is a polynomial $p_c(\tau)$ ($\approx \tau(\tau - 1)/2$) independent from $x' = M_i$ such that:

1. If N_c on M_i does not halt in $p_c(\tau)$ steps (i.e., N_c on M_i has no accepting path of length $\leq p_c(\tau)$), then (**), and
2. N_c on M_i does not halt in $p_c(\tau)$ steps iff M^* accepts $\langle N_c, M_i, 1^{p_c(\tau)} \rangle$ in $p_{N_c, M_i}(p_c(\tau))$ steps.

Thus, by running M^* on $\langle N_c, M_i, 1^{p_c(\tau)} \rangle$, we can guarantee (**) (if it is indeed possible) in $p(\tau)$ steps for some polynomial determined by M_i .

Then the running time of M_{opt} satisfies condition (1) of Def. 2.2, which contradicts that **coBHP** has superpolynomial speedup. ■

Interestingly, each problem identified by Chen and Flum [5] as having the same complexity as **coBHP** under fixed parameter tractable reductions, such as the set of arithmetic statements ϕ with no proof of fewer than t steps, is also the resource-bounded version of a non-c.e. language.

3. Conclusion

Their result and the parallel between Fact 2.1 and condition (*) suggest a correspondence between known facts in computability theory and hypotheses in complexity theory. As another example, Gödel demonstrated speedup in the length of proofs of arithmetic statements [8], and a corresponding conjecture in proof complexity is that there is speedup in the length of proofs of tautologies (no p -optimal propositional proof system).

This correspondence serves several purposes. First, it suggests statements such as (*) which are interesting in themselves. Second, the correspondence may be interpreted as (weak) evidence that there are superpolynomial speedups for accepting **coNP**-complete languages and for proof length for propositional proof systems as has been conjectured [12]. Finally, we suspect that the validity of the corresponding statements such as Fact 2.1 and (*) are closely linked.⁶ Resource-bounded versions of noncomputable problems may misbehave by failing to have an optimal algorithm or proof system, just as their noncomputable counterparts misbehave by failing to have any algorithm or proof system at all.

To pursue this linkage, we can define a version of (*) which like Fact 2.1 is constructive. For M accepting **coBHP**, let E be the set of non-halting $\langle N, x \rangle$ for which $f(t)$ is polynomially bounded. Suppose there is an M_E which accepts E . Then a constructive version of (*) is:⁷

Conjecture 3.1. *The $\langle N', x' \rangle$ for M_E predicted by Fact 2.1 satisfies (*) for M .*

The condition (*) was motivated by our suspicion that the existence of a polynomial time M accepting **coBHP** would violate the information constraint imposed by the noncomputability of the halting problem. More precisely:

⁶Fact 2.1 implies a very weak, model dependent speedup for **coBHP** (for details see [16]). For M accepting **coBHP**, let S_M be the set of non-halting $\langle N, x \rangle$ such that M accepts $\langle N, x, 1^\infty \rangle$ in finite time, where the encoding is such that M does not necessarily read the full input. By Fact 2.1, S_M does not include some non-halting $\langle N', x' \rangle$. Then for any M , there exists M' such that $S_{M'} = S_M \cup \{\langle N', x' \rangle\}$ is a strictly larger set than S_M . This M' avoids reading the full input in more cases than does M . This line of argument also holds for **coBHP** with N deterministic, and does not hinge on the fact that **coBHP** is **NP**-complete.

⁷Joseph and Young [10] and Wang [19] define p -productive languages where the productive function yields a single problematic input, whereas in Conjecture 3.1 an infinite family of inputs is produced.

Conjecture 3.2. *If there exists $M \in \mathbf{P}$ accepting coBHP, then M can be modified to accept all non-halting $\langle N, x \rangle$.*

More broadly, we wonder whether the obstacle to the existence of a TM M which acts contrary to various widely believed complexity hypotheses is that M could be modified to perform a related task known to be noncomputable. For instance, it is curious that arithmetic is undecidable only if it incorporates multiplication, and that this fact has not been used to say anything about the complexity of integer multiplication (which may have a slight speedup) or the inverse operation of factorization.

References

- [1] D. Bernstein, Multidigit multiplication for mathematicians, mimeograph, <http://cr.yp.to/papers/m3.pdf> (2001).
- [2] M. Blum, A machine-independent theory of the complexity of recursive functions, *J. ACM* 14 (1967) 322–36.
- [3] M. Blum, 2007. Private communication.
- [4] Y. Chen, J. Flum, A logic for PTIME and a parameterized halting problem, in: *LICS '09: Proceedings of the 2009 24th Annual IEEE Symposium on Logic In Computer Science*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 397–406.
- [5] Y. Chen, J. Flum, On slicewise monotone parameterized problems and optimal proof systems for TAUT, in: A. Dawar, H. Veith (Eds.), *Computer Science Logic*, volume 6247 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, pp. 200–214.
- [6] S.A. Cook, R.A. Reckhow, The relative efficiency of propositional proof systems, *J. Symb. Log.* 44 (1979) 36–50.
- [7] D. Coppersmith, S. Winograd, On the asymptotic complexity of matrix multiplication, *SIAM J. Comput.* 11 (1982) 472–92.
- [8] K. Gödel, Über die lange von beweisen, *Ergebnisse eines Mathematischen Kolloquiums* 7 (1936) 23–24.

- [9] E.A. Hirsch, D. Itsykson, On optimal heuristic randomized semidecision procedures, with application to proof complexity, in: J.Y. Marion, T. Schwentick (Eds.), STACS, volume 5 of *LIPICs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 453–464.
- [10] D. Joseph, P. Young, Some remarks on witness functions for nonpolynomial and noncomplete sets in NP, *Theor. Comput. Sci.* 39 (1985) 225–237.
- [11] J. Köbler, J. Messner, Is the standard proof system for SAT P-optimal?, in: S. Kapoor, S. Prasad (Eds.), FSTTCS, volume 1974 of *Lecture Notes in Computer Science*, Springer, 2000, pp. 361–372.
- [12] J. Krajíček, P. Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations, *J. Symb. Log.* 54 (1989) 1063–79.
- [13] L.A. Levin, Universal sequential search problems, *Problems of Information Transmission* 9 (1973) 265–66.
- [14] A.R. Meyer, P.C. Fischer, Computational speed-up by effective operators, *J. Symb. Log.* 37 (1972) 55–68.
- [15] H. Monroe, Are there natural problems with speedup?, *Bulletin of the European Association for Theoretical Computer Science* 94 (2008) 212–20.
- [16] H. Monroe, Speedup for natural problems and $\text{coNP} \stackrel{?}{=} \text{NP}$, *Electronic Colloquium on Computational Complexity (ECCC)* (2009).
- [17] V. Pan, *How to Multiply Matrices Faster*, Springer-Verlag, New York, NY, 1984.
- [18] C.P. Schnorr, G. Stumpf, A characterization of complexity sequences, *Zeitschr. für Math. Logik und Grundlagen der Mathematik* 21 (1975) 47–56.
- [19] J. Wang, P-productivity and polynomial time approximations, in: *Structure in Complexity Theory Conference* (1990), pp. 254–265.