On the Sparse Set Conjecture for Sets with Low Density

Harry Buhrman*[1] and Montserrat Hermo**[2]

[1] CWI PO Box 94079, 1090 GB Amsterdam, The Netherlands,
buhrman@cwi.nl

[2] Universidad del Pais Vasco, Facultad de Informatica,
Apdo. 649 , 20080, San Sebastian, Spain,
jiphehum@si.ehu.es

Abstract. We study the sparse set conjecture for sets with low density. The sparse set conjecture states that $P = NP$ if and only if there exists a sparse Turing hard set for NP. In this paper we study a weaker variant of the conjecture. We are interested in the consequences of NP having Turing hard sets of density $f(n)$, for (unbounded) functions $f(n)$, that are sub-polynomial, for example $\log(n)$. We establish a connection between Turing hard sets for NP with density $f(n)$ and bounded nondeterminism: We prove that if NP has a Turing hard set of density $f(n)$, then satisfiability is computable in polynomial time with $O(\log(n) * f(n^c))$ many nondeterministic bits for some constant c. As a consequence of the proof technique we obtain absolute results about the density of Turing hard sets for EXP. We show that no Turing hard set for EXP can have sub-polynomial density. On the other hand we show that these results are optimal w.r.t. relativizing computations. For unbounded functions $f(n)$, there exists an oracle relative to which NP has a $f(n)$ dense Turing hard tally set but still $P \neq NP$.

1 Introduction

The density of NP -complete and hard sets has been an early object of study and starts with the seminal paper of Berman and Hartmanis [BH77]. In that paper roughly two lines of research have been initiated: the density of $\leq_m^p$ -complete sets and the density of $\leq_T^p$ -hard sets for NP and other complexity classes.

The study of the $\leq_m^p$ -complete sets for NP becomes apparent in relation with the isomorphism conjecture [BH77]. The conjecture implies that all NP - complete sets are exponentially dense, since SAT, the well known NP -complete

* Part of this research was done while visiting the Univ. Politècnica de Catalunya in Barcelona. Partially suported by the Dutch foundation for scientific research (NWO) through NFI Project ALADDIN, under contract number NF 62-376 and a TALENT stipend.

** Part of this research was done while visiting the Univ. Politècnica de Catalunya in Barcelona.

set, itself is of exponential density. Along these lines it was Mahaney who showed that NP -complete sets can not be polynomially dense unless $P = NP$.

The study of the $\leq_T^p$ -hard sets for NP is motivated by the equivalence between sets that are $\leq_T^p$ -reducible to a sparse set, sets that have polynomial size circuits [BH77] and sets that can be recognized in polynomial time with the additional help of a polynomial amount of advice ($P/poly$) [Pip79]. Hence if there exists a $\leq_T^p$ -hard sets for NP that is sparse, there exists a polynomial time algorithm for SAT that needs the help of a small (polynomial size) table. For practical purposes this would mean that one only had to compute this small table once (for inputs of a certain length) and that from then on NP would equal P for all inputs of this length. Karp and Lipton however showed that the existence of sparse $\leq_T^p$ -hard sets for NP implies an unlikely collapse of the Polynomial Time Hierarchy to its second level [KL80].

Many efforts have been put into improving the Karp and Lipton result to $P = NP$. This improvement also goes by the name "sparse set conjecture". An important step towards this conjecture was obtained by Ogiwara and Watanabe, who showed that indeed the stronger consequence, i.e. $P = NP$, can be obtained if $\leq_{btt}^p$ -reductions are used instead of $\leq_T^p$ -reductions [OW91].

Not only does the sparse set conjecture imply a better understanding of the structure of NP and NP -hard sets, also does it settle the blatant unability to prove that EXP does not have polynomial size circuits. Best known upper bounds on this can be found in [Kan82].

Attempts to prove the sparse set conjecture usually result in studying stronger than Turing reducibility types to sparse sets [OW91, HL94, OL91, AHH+93]. In this paper we follow a different line of attack. Instead of strengthening the reduction type we study the most general Turing reduction and vary the density of the set reduced to. We are in particular interested in the consequences of the existence of $\leq_T^p$ -hard sets for NP that have *smaller* density than sparse sets. In [HOT] a similar approach was taken with respect to log-space-bounded reductions. The analogous question for sets with bigger than polynomial, i.e. super-polynomial, density has been addressed in [BH92].

We study the consequences of the existence of $\leq_T^p$ -hard sets for NP, that contain only $f(n)$ strings of size less than or equal to n, for $f(n)$ a unbounded function that is strictly smaller than any polynomial. We show that there is a link between classes of bounded nondeterminism [KF80, DT90], and NP having hard sets with low density. We prove that under the assumption that NP has a $\leq_T^p$ -hard set with density $f(n)$, SAT can be computed with only $O(\log(n) * f(n^c))$ bits of nondeterminism, for some constant c. Taking for $f(n)$ for example $\log(n)$ results in a collapse of NP to the second level of the Beta Hierarchy [DT90]. This on its turn implies for example that $EXP = NEXP$. Note P can be characterized as the class of sets that are recognized by nondeterministic polynomial time machines that use $O(\log(n))$ bits of nondeterminism.

On the other hand we show that this result is optimal with respect to relativizing computations, even if we consider tally sets. We prove that there exists an oracle relative to which NP does have Turing hard tally sets with $f(n)$ density but $P \neq NP$, for $f(n)$ an unbounded function.

As an application of the developed proof technique we establish absolute results concerning the density of $\leq_T^p$ -hard sets for EXP. We prove that Turing hard sets for EXP can not be of sub-polynomial density. This is in some sense optimal, since improvement to polynomial density, would show that EXP does not have polynomial size circuits, and Wilson [Wil85] showed the existence of an oracle where EXP does have polynomial size circuits.

The main results in this paper are:

- We establish a connection between classes of bounded nondeterminism and NP having Turing hard sets with low density.
- We prove that if NP has a Turing hard set with density $f(n)$, for any unbounded function $f(n)$, then SAT can be computed in polynomial time with the use of $O(\log(n) * f(n^c))$ many nondeterministic bits, for c some constant and hence $EXP = NEXP$.
- We show that these results are optimal with respect to relativized computations: there exists an oracle relative to which NP has a Turing hard tally set with density $f(n)$, but $P \neq NP$.
- As a consequence of the developed proof technique we show that Turing hard sets for EXP can not have sub-polynomial density.

The results suggest that it is probably hard to prove the sparse set conjecture even if we consider sets with arbitrary low density. On the other hand this line of research might give a handle on proving the actual sparse set conjecture. It seems more doable to work on non-relativizing proof techniques for proving the sparse set conjecture, for sets with low density than for sparse sets.

2 Preliminaries

We assume the reader familiar with standard notions in structural complexity theory, as are defined e.g. in [BDG88]. We will be using (non)deterministic polynomial time oracle Turing machines. Let M be a (non)deterministic polynomial time oracle Turing machine. We will denote $Q(M, x, A)$ as the set of queries M makes on input x with oracle A. Note that if M is a deterministic machine then $\|Q(M, x, A)\|$ is bounded by a polynomial in the length of x.

Apart from SAT, the well known NP -complete set, we will be using the set K^A as well. Where K^A is defined as follows:

$$K^A = \{<i, x, 0^{|x|^i}> \mid M_i^A(x) \text{ accepts x within } |x|^i \text{ steps } \}$$

We will be considering efficient reductions of complete sets for various classes to sets of different subpolynomial densities. The following definition specifies our measure of a set's density.

Definition 1. Let f be a nondecreasing function, $f : I\!N \rightarrow I\!N$. A set $S \subseteq \Sigma^*$ is $f(n)$ dense or has $f(n)$ density if $\|S^{\leq n}\| < f(n)$ for all n. If F is a class of functions, we say S is F-dense if S is f dense for some $f \in F$.

S is said to be *sparse* if S is P-dense. Where P is the class of all polynomials. S is said to be of *sub-polynomial density* if S is F-dense, where F is a class of functions such that $f \in F$ iff $\forall\epsilon\exists n_0\forall n > n_0$: $f(n) < n^\epsilon$. We will call such a function *sub-polynomial.*

We will be using classes that are defined by limiting the number of nondeterministic moves of a nondeterministic polynomial time Turing machine [KF80, DT90].

Definition 2. [DT90] For any function $f : \mathbb{N} \rightarrow \mathbb{N}$ let

$$\beta_f = \{L \mid \exists A \in P, \exists c, \forall x : x \in L \Leftrightarrow (\exists y, |y| \leq c * f(|x|) \text{ and } <x,y> \in A)\}$$

Will will also use the notation $\beta_f^{\beta_f}$ for the class of sets recognizable by a polynomial time oracle Turing machine that uses at most $f(n)$ nondeterministic moves and has a set in β_f as oracle.

3 Hard sets for NP and Bounded Nondeterminism

In this section we study the consequences of *NP* having Turing hard sets with low density. We are interested in sets with sub-polynomial density.

Let us consider an example of such a set. Let $f(n)$ be $\log(n)$. Consider the assumption that *NP* has a Turing hard tally set T of density $\log(n)$. This means that SAT $\leq_T^p T$, say in time n^c. The following counting argument, together with a nowadays standard technique, yield that SAT is computable in time $DTIME(c^2 * n^{\log(n)})$, for some constant c.

First lets count how many different tally sets $T^{\leq n^c}$ of density $\log(n)$ there exist. Each $T^{\leq n^c}$ may contain at most $c * \log(n)$ strings, that can be placed at n^c many different positions, hence the number of different tally sets up to length n^c is bounded by:

$$\binom{n^c}{c\log(n)} \leq n^{c^2 * \log(n)}$$

Next we will use the fact that it is possible to compute in polynomial time relative to SAT, for any satisfying formula ϕ, using the disjunctive selfreducibility of SAT, a satisfying assignment for ϕ. This property is also called Search Reduces to Decision or Functional Selfreducibility [BD76, BBFG91, NOS93]. Since SAT reduces to some tally set with $\log(n)$ density, it is reducible to one among the $n^{c^2 * \log(n)}$ many different ones. Consider the following algorithm. On input ϕ cycle through all the possible tally sets of $\log(n)$ density. For each one try to compute a satisfying assignment for ϕ and accept if and only if one is found . It is clear that a satisfying assignment will be found this way if and only if $\phi \in$ SAT. Furthermore this procedure runs in time $n^{c^2 * \log(n)}$.

The problem with the previous approach is that it does not work for sets over $\{0, 1\}$. The number of possible sets with density $\log(n)$ is only bounded by $2^{n^{c * \log(n)}}$ and hence will yield that SAT is computable in time $2^{n^{c * \log(n)}}$, but this is not very good since SAT is computable in time 2^n. The following theorem shows that linking the problem to bounded nondeterminism yields even better results than the above approach for non tally sets. We will show that for any function $f(n)$ the assumption that NP has a Turing hard set with density $f(n)$ implies that SAT is computable in polynomial time with $O(\log(n) * f(n^c))$ many nondeterministic bits, for some constant c.

Theorem 3. *Let $f(n)$ be any fully time constructible function. If there exists a set S with density $f(n)$ that is Turing hard for NP then SAT is computable in polynomial time with $O(\log(n) * f(n^c))$ many nondeterministic bits, for some constant c.*

Proof. (Sketch)We have to show that under the assumption that SAT reduces to a set S with density $f(n)$ we can construct a polynomial time algorithm using $O(\log(n) * f(n^c))$ many nondeterministic bits, that decides SAT. Assume that SAT $\leq_T^p S$ via a machine M_c, that runs in time n^c and that M_a witnesses the fact that SAT has Search Reducing to Decision. Simulate the machine M_a, that generates a satisfying assignment relative to SAT on input ϕ. Every time M_a makes a query to SAT simulate the Turing reduction M_c, from A to S, on this query. Every time M_c makes a query q to S either assume that q is out of S or guess that q is in S. Make sure that at most $f(n^c)$ times a query is guessed to be in S and that the decision about q is consistent with previously made decisions about q. Accept if and only if a satisfying assignment for ϕ is found. It should be clear that this construction runs in polynomial time. From this it follows that at most $O(\log(n) * f(n^c))$ many nondeterministic bits are used. The following describes in more detail the above algorithm:

input ϕ
$n := |\phi|$
Guess a set $Positions$ with $\leq f(n^c)$ pointers of size $\leq \log(n^{c+1})$
$Queries := \emptyset$
$Count := 0$
$z := \phi$
  while there is at least one variable in z
    let x be the first variable in z
    $z' := z|_{x:=0}$
    while Simulation of M_c on z' not ended **do**
      Simulate M_c on z' until it queries q
      $Count := Count + 1$
      if $q \in Queries$ **then** continue simulation in YES state
      else if $Count \in Positions$ **then**
        $Queries := Queries \bigcup \{q\}$
        continue simulation in YES state
        else continue simulation in NO state
    end while
    if M_c accepted z' **then** $z := z'$
    else $z = z|_{x:=1}$

end while
if z simplifies to "true" **then** accept **else** reject

Where $z|_x := i$ for $i = 0, 1$ means the formula obtained by substituting i for every occurrence of x in z.

The above Theorem shows that although a set S over $\{0,1\}$ with density $f(n)$ (for $f(n)$ small) contains in some sense more information than a tally set T of the same density, a polynomial time algorithm is not able to extract this information out of S.

Plugging in explicit values for $f(n)$ yields the following corollary:

Corollary 4. *If there exists a set S with density $(\log(n))^k$, that is Turing hard for NP, then:*

1. *$NP = \beta_{polylog}$.*
2. *$EXP = NEXP$.*

Proof. Use Theorem 3 together with standard padding arguments.

Another consequence of the proof technique of Theorem 3 is that it yields absolute results about the density of Turing hard sets for EXP. To our knowledge this is the first result concerning the density of $\leq_T^p$ -hard sets for EXP.

Theorem 5. *There do not exist Turing hard sets for EXP that have sub-polynomial density.*

Proof. (Sketch) Assume that there exists a $f(n)$ dense Turing hard set for EXP for some sub-polynomial function f. From this we can conclude [KL80] that $EXP = \Sigma_2^p$. We will see that by extending Theorem 3 we will be able to show that Σ_2^p is computable in time $2^{O(\log(n)*f(n^{O(1)}))}$. Then we will have a contradiction with the hierarchy Theorems for deterministic time, since EXP is not computable in sub-exponential deterministic time. In order to extend Theorem 3 we will make use of the fact that all $\leq_T^p$ -complete sets for Σ_2^p are functional selfreducible [BD76]. Functional selfreducibility is the natural generalization of search reducing to decision for other levels of the Polynomial Time Hierarchy. Consider any set A in Σ_2^p. By definition there exists a polynomial time computable relation R_A such that:

$$x \in A \Leftrightarrow \exists y \forall z : R_A(x, y, z)$$

A set $A \in \Sigma_2^p$ is functional selfreducible if there exists a polynomial time procedure that can compute, relative to A, for all $x \in A$ a y_0 such that $\forall z$: $R_A(x, y_0, z)$. For all strings not in A such a y_0 does not exists. Borodin and Demers showed that this is always the case if A is $\leq_T^p$ -complete for Σ_2^p [BD76]. We will see that under the assumption of the existence of a $f(n)$ dense hard set for EXP any complete set A for Σ_2^p can be recognized in $\beta_{g(n)}^{\beta_{g(n)}}$, for $g(n) \in O(\log(n) * f(n^{O(1)}))$.

Let A be $\leq_T^p$ -complete for Σ_2^p, M_f be the procedure that witnesses that A is functional selfreducible and M_c be the machine witnessing that $A \leq_T^p S$. Simulate M_f on some input x and every time a query to A is made simulate the reduction M_c, from A to S, on this query . Again, as in the proof of Theorem 3, if M_c makes a query q to S assume either that q is not in S or guess that the query is in the set, making sure that no more than $f(n^c)$ many yes guesses are made and that the decisions are consistent with previously made decisions about q. After this computation some y' has been computed. If nothing is computed, because y_0 does not exists or because some wrong sequence of guesses is used, reject. Next the algorithm checks whether the computed y' indeed satisfies the property that $\forall z : R_A(x, y', z)$. This can be done by one oracle call to SAT. Since SAT $\in \Sigma_2^p$ it follows using Theorem 3 that SAT $\in \beta_{O(\log(n)*f(n^d))}$ for some d. Hence $A \in \beta_{g(n)}^{\beta_{g(n)}}$ and can be decided in deterministic sub-exponential time.

Note that any improvement to the previous theorem will settle the big open question whether EXP has polynomial size circuits. Also does the theorem hint at the possibility that a much weaker form of the sparse set conjecture[3], might settle the circuit issue for EXP.

4 Relativized Optimality

In this section we show that Theorem 3 is optimal with respect to relativized computations.

Theorem 6. *Let $f(n)$ be an unbounded function. There exists an oracle A and a tally set T of density $f(n)$ such that:*

1. $K^A \leq_T^{p,A} T$ *and*
2. $P^A \neq NP^A$.

Proof. (Sketch) W.l.o.g. we assume that $f(n) < n$ and monotone. The construction of the set A consists out of two parts. One to satisfy requirement 1 and the other part to satisfy 2. The set A will be the marked union of QBF, a *PSPACE* $\leq_m^p$ -complete set, and a set B (i.e. $A =$QBF$\oplus B$). Requirement 2 will be satisfied by showing that the following test language L is in NP^A but not in P^A:

$$L = \{0^n \mid \exists x \in B \text{ and } |x| = n\}$$

Note that for all B, L is in NP^A. We will construct B in such a way that $L \notin P^A$. In order to do this we need a sequence of strings to diagonalize over. Let $\{a_i\}_{i=0}^{\infty}$ be a sequence of natural numbers such that the following things hold:

[3] We mean here the following weaker variant: SAT is computable in polynomial time with significantly less than a linear number of nondeterministic bits if and only if there exists a sparse Turing hard set for NP.

1. $f(a_i) \geq 2 * f(a_{i-1})$,
2. $f(a_{i-1}) * a_{i-1} < a_i$,
3. $a_i^{f(a_i)} > a_i^i + i$, and
4. a_0 is the first n such that $f(n) \geq 2$.

Since $f(n)$ is unbounded it is not hard to see that $\{a_i\}_{i=0}^{\infty}$ exists and is infinite. In the following let $A = \text{QBF} \oplus B$ and let $M_1, M_2 \ldots$ be an enumeration of polynomial time oracle Turing machines such that M_i runs in time $n^i + i$. The construction of B goes as follows:

stage n:

Let $y_{a_i} = x_1 x_2 \ldots x_k 00 \ldots 0$ be a string such that:

1. $k = \log(a_i) * f(a_i)$,
2. $x_j \in \{0, 1\}$ $(1 \leq j \leq k)$,
3. $|y_{a_i}| = a_i$, and
4. $y_{a_i} \notin Q(M_i, 0^{a_i}, A)$.

Put y_{a_i} into B if and only if $M_i^A(0^{a_i})$ rejects.

end of stage n

From the definition of $\{a_i\}_{i=0}^{\infty}$, the fact that M_i runs in time $n^i + i$, and that $f(n)$ is unbounded it follows that y_{a_i} exists and that $L \notin P^A$. The construction of B satisfies requirement 2. In order to satisfy requirement 1 we need to code B into a $f(n)$ dense tally set. We will show that we can code B into a $2 * f(n)$ dense tally set T, which is sufficient[4] to satisfy requirement 1. First observe the following. If we put, for all n, $f(a_n)$ strings into T at the interval $0^{a_n}, \ldots, 0^{a_n * f(a_n)}$ then T will have density $2 * f(n)$. The observation yields that we have $f(a_n)$ strings to code y_{a_n} into T. To do this divide the interval $0^{a_n}, \ldots, 0^{a_n * f(a_n)}$ into $f(a_n)$ intervals I_k of length a_n such that $I_k = 0^{k * a_n}, \ldots, 0^{((k+1) * a_n - 1)} (1 \leq k < f(a_n))$. Next if we place one string t_k in each I_k we can interpret $t_1 t_1 \ldots t_{f(a_n)}$ as a $f(a_n)$ digit number base a_n. Recall that $y_{a_n} = x_1 x_2 \ldots x_k 00 \ldots 0$ where $k = \log(a_n) * f(a_n)$. The only part of y_{a_n} that matters is $x_1 x_2 \ldots x_k$. Since we can express $x_1 x_2 \ldots x_k$ as a $f(a_n)$ digit number base a_n there is room to code $x_1 x_2 \ldots x_k$ and hence y_{a_n} into T.

This coding argument shows that we can code B into T. To see that $K^A \leq_T^{p,A} T$ we do the following reduction. Let M_j be some nondeterministic polynomial time oracle machine that accepts K^A. In order to know whether $M_j^A(x)$ accepts or rejects, the reduction firsts recovers B up to length $|x|^j$, using T. Next it queries QBF whether $M_j(x)$ with QBF and B up to length $|x|^j$ accepts or rejects.

Note that the above construction actually proves something stronger. It shows that there exists an oracle A such that NP^A has a Turing hard tally set of $f(n)$ density and even $P^A \neq \beta^A_{\log(n) * f(n)}$.

[4] This actually only shows that for each unbounded function $f(n)$ there exists a $2 * f(n)$ dense tally set with the desired properties. However starting out with a function $f'(n) = f(n)/2$ will satisfy requirement 1.

5 Conclusions

The oracle result (Theorem 6) shows that the weaker form of the sparse set conjecture needs nonrelativizing proof techniques to be proven. On the other hand it seems more doable to develop non-relativizing techniques for the weaker form of the conjecture than for the actual conjecture.

Another interesting point that comes out of the oracle construction is that the idea behind the coding can be used to show that the class of sets that Turing reduce to a tally set with polylog[5] density and the nonuniform advice class *Full-P/polylog* [Ko87, BHM92] are equal. It seems therefore natural to study the structure of these advice classes in greater detail.

Theorem 5 suggests that the weaker form of the sparse set conjecture, i.e $NP = \beta_f(n)$ iff exists a Turing hard sparse set for NP for some subpolynomial function $f(n)$, implies that EXP is not contained in $P/poly$. It would be interesting to prove this.

Acknowledgements

We would like to thank José Balcázar and Ricard Gavaldà for listening to us with great patience and suggesting interesting ideas. Special thanks are due to Jacobo Torán for pointing out the importance of limited nondeterminism.

References

[AHH+93] V. Arvind, Y. Han, L. Hemachandra, J. Koebler, A. Lozano, M. Mundhenk, M. Ogiwara, U. Schöning, R. Silvestri, and T. Thierauf. Reductions to sets of low information content. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory*, pages 1–46. Cambridge University Press, December 1993.

[BBFG91] R. Beigel, M. Bellare, J. Feigenbaum, and S. Goldwasser. Languages that are easier to verify than their proofs. In *Proc. 32nd IEEE Symposium on Foundations of Computer Science*, pages 19–28, 1991.

[BD76] A. Borodin and A. Demers. Some comments on functional self-reducibility and the NP hierarchy. Technical Report TR76-284, Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853, 1976.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, 1988.

[BH77] L. Berman and H. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6:305–322, 1977.

[BH92] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In R. Shyamasundar, editor, *Proc. 12th Conference on the Foundations of Software Technology & Theoretical Computerscience, Lecture Notes in Computer Science*, pages 116–127. Springer Verlag, 1992.

[5] Polylog stands for the class of functions $\{f \mid \exists i, f(n) = (\log(n))^i\}$.

[BHM92] J.L. Balcázar, M. Hermo, and E. Mayordomo. Characterizations of logarithmic advice complexity classes. In *Algorithms, Software, Architecture: Information Procesing 92*, volume 1, pages 315–321. Elsevier, 1992.

[DT90] J. Díaz and J. Torán. Classes of bounded nondeterminism. *Math. Systems Theory*, 23:21–32, 1990.

[HL94] S. Homer and L. Longpré. On reductions of NP sets to sparse sets. *J. Computer and System Sciences*, 48:324–336, 1994.

[HOT] L. Hemaspaandra, M. Ogiwara, and S. Toda. Space-efficient recognition of sparse self-reducible languages. *Computational Complexity*. In press.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1–3):40–56, October/November/December 1982.

[KF80] C. Kintala and P. Fisher. Refining nondeterminism in relativized polynomial-time bounded computations. *SIAM J. Comput.*, 9(1):46–53, 1980.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[Ko87] K. Ko. On helping by robust oracle machines that take advice. *Theoretical Computer Science*, 52:15–36, 1987.

[NOS93] A.V. Naik, M. Ogiwara, and A.L. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. In *Proc. Structure in Complexity Theory 8th annual conference*, pages 52–64, San Diego, California, 1993. IEEE computer society press.

[OL91] M. Ogiwara and A. Lozano. On one query self-reducible sets. In *Proc. Structure in Complexity Theory 6th annual conference*, pages 139–151, Chicago, Ill., 1991. IEEE Computer Society Press.

[OW91] M. Ogiwara and O. Watanabe. On polynomial time bounded truth-table reducibility of NP sets to sparse sets. *SIAM J. Comput.*, 20:471–483, 1991.

[Pip79] N. Pippenger. On simultaneous resource bounds. In *Proc. 20th IEEE Symposium on Foundations of Computer Science*, pages 307–311, 1979.

[Wil85] C.B. Wilson. Relativized circuit complexity. *J. Comput. System Sci.*, 31:169–181, 1985.