J Comb Optim
DOI 10.1007/s10878-010-9327-5

Separating NE from some nonuniform nondeterministic complexity classes

Bin Fu · Angsheng Li · Liyu Zhang

Abstract We investigate the question whether NE can be separated from the reduction closures of tally sets, sparse sets and NP. We show that (1) NE $\not\subseteq R^{\mathrm{NP}}_{n^{o(1)}-T}(\mathrm{TALLY})$; (2) NE $\not\subseteq R_m^{SN}(\mathrm{SPARSE})$; (3) NEXP $\not\subseteq \mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^k$ for all $k \geq 1$; and (4) NE $\not\subseteq \mathrm{P}_{btt}(\mathrm{NP} \oplus \mathrm{SPARSE})$. Result (3) extends a previous result by Mocas to nonuniform reductions. We also investigate how different an NE-hard set is from an NP-set. We show that for any NP subset A of a many-one-hard set H for NE, there exists another NP subset A' of H such that $A' \supseteq A$ and $A' - A$ is not of sub-exponential density.

Keywords NE · NEXP · Nonuniform complexity class · Separation · Complexity

1 Introduction

Separating the complexity classes has been one of the central problems in complexity theory. Approximation is widely studied method to use low resource computa-

A preliminary version of this work appeared as "Separating NE from Some Nonuniform Nondeterministic Complexity Classes" in the Proceedings of the 15th International Computing and Combinatorics Conference (COCOON'2009), Lecture Notes in Computer Science, vol. 5609, pp. 486–495, 2009.

B. Fu (✉)
Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA
e-mail: binfu@cs.panam.edu

A. Li
Institute of Software, Chinese Academy of Sciences, Beijing, China
e-mail: angsheng@gcl.iscas.ac.cn

L. Zhang
Department of Computer and Information Sciences, University of Texas at Brownsville, Brownsville, TX 78520, USA
e-mail: liyu.zhang@utb.edu

Published online: 22 April 2010

tion method to deal with some high resource computation. We would like to see how a hard problem in NE can be approximated by a problem in NP. The difference between NE and NP has not been fully solved. One of the most interesting problems between them is to separate NE from $P_T(NP)$. This paper continues a line of research that tries to separate nondeterministic complexity classes in a stronger sense, i.e., separating nondeterministic complexity classes from the *reduction closure* of classes with lower complexity. We focus on the class NE of nondeterministically exponential-time computable sets. Two most interesting but long standing open problems regarding NE are whether every NE-complete set is polynomial-time Turing reducible to an NP set and whether it is polynomial-time Turing reducible to a sparse set. The latter question is equivalent to whether every NE-complete set has polynomial-size circuits, since a set is polynomial-time Turing reducible to a sparse set if and only if it has polynomial-size circuits (Berman and Hartmanis 1977). We show results that generalize and/or improve previous results regarding these questions and help to better understand them. In complexity theory, a sparse set is a set with polynomially bounded density. Whether sparse sets are hard for complexity classes is one of the central problems in complexity theory (Mahaney 1982; Ogiwara and Watanabe 1991; Karp and Lipton 1980; Cai and Sivakumar 1999). In particular, Mahaney (1982) showed that sparse sets cannot be many-one complete for NP unless $P = NP$. Sparse sets play an important role in the study of complexity theory (Mahaney 1982; Ogiwara and Watanabe 1991; Karp and Lipton 1980; Cai and Sivakumar 1999). Ogihara and Tantau (2004) characterized sparse sets and p-selective sets as “sets with low information content” as both sparse sets and p-selective sets become tractable when a small amount of addition information, i.e., advice bits, is available. They studied whether problems of NP complexity can be reduced in polynomial-time to sparse sets via various reductions. They proved that several important problems inside NP including the Satisfiability problem and the Graph Isomorphism problem cannot be reduced to sparse sets via certain restrictions of the general polynomial-time Turing reduction. In Sect. 3 we study the question whether sparse sets can be hard for NE under reductions that are weaker than the polynomial-time Turing reductions. We prove that no NE-hard set can be reducible to sparse sets via the *strong nondeterministic polynomial-time many-one reduction*. For a special case of sparse sets, tally sets, we strengthen the result to the *nondeterministic polynomial-time Turing reductions* that make at most $n^{o(1)}$ many queries. These are the main results of this paper. They can be viewed as a step towards understanding the information content of high-complex sets such as NE-hard sets. Note that generalizing these results to polynomial-time Turing reductions is hard since already the deterministic polynomial-time Turing reduction closure of spare sets as well as that of p-selective sets equals P/*poly* (Hemaspaandra and Torenvliet 2003), and it is not even known whether $NE \not\subseteq P/poly$.

We present a new result on the aforementioned long standing open question whether every NE set is polynomial-time Turing-reducible to a NP set. Fu et al. (1992) first tackled this problem and showed that $NE \not\subseteq P_{n^{o(1)}-T}(NP)$. Their result was later improved by Mocas (1996) to $NEXP \not\subseteq P_{n^c-T}(NP)$ for any constant $c > 0$. Mocas’s result is optimal with respect to relativizable proofs, as Buhrman and Torenvliet (1994) constructed an oracle relative to which $NEXP = P^{NP}$. In this paper, we extend Mocas's result to nonuniform polynomial-time Turing reductions that uses a fixed polynomial number of advice bits. More precisely, we show that NEXP $\not\subseteq$ P$_{n^k-T}$(NP)$/n^{k'}$ for any constant $k, k' > 0$. Since it is easy to show for any $k > 0$ that $P_{n^k-T}(\text{NP} \oplus \text{P-Sel}) \subseteq P_{n^k-T}(\text{NP})/n^k$, where P-Sel denotes the class of p-selective sets, we obtain as a corollary that NEXP $\not\subseteq$ P$_{n^k-T}$(NP $\oplus$ P-Sel).

We investigate a different but related question. We study the question of how different a hard problem in NE is from a problem in NP. One way to measure the difference between sets is by using the notion of *closeness* introduced by Yesha (1983). We say two sets are f-close if the density of their symmetric difference if bounded by $f(n)$. The closeness to NP-hard sets were further studied by Fu (1993) and Ogiwara (1991). We show that for every $\leq_m^P$-complete set H for NE and every NP-set $A \subseteq H$, there exists another NP-set $A' \subseteq H$ such that $A \subseteq A'$ and A' is not subexponential-close to A. For coNE-complete sets we show a stronger result. We show that for every $\leq_m^p$-complete set H for coNE and every NP-set $A \subseteq H$, there exists another NP-set $A' \subseteq H$ such that $A \cap A' = \emptyset$ and A' is exponentially dense. Finally, we also show that NE $\not\subseteq$ P$_{btt}$(NP $\oplus$ SPARSE).

2 Notations

We use standard notations (Homer and Selman 2001; Hemaspaandra and Ogihara 2002) in structural complexity. All the languages throughout the paper are over the alphabet $\Sigma = \{0, 1\}$. For a string x, $|x|$ is the length of x. For a finite set A, $||A||$ is the number of elements in A. We use Σ^n to denote the set of all strings of length n and for any language L, $L^{=n} = L^n = L \cap \Sigma^n$. We fix a pairing function $\langle\cdot\rangle$ such that for every $u, v \in \Sigma^*$, $|\langle u, v\rangle| = 2(|u| + |v|)$. For a function $f(n) : N \to N$, f is *exponential* if for some constant $c > 0$, $f(n) \geq 2^{n^c}$ for all large n, and is *subexponential* if for every constant $c > 0$, $f(n) \leq 2^{n^c}$ for all large n. A language L is *exponentially dense* if there exists a constant $c > 0$ such that $||L^{\leq n}|| \geq 2^{n^c}$ for all large n. Let Density($d(n)$) be the class of languages L such that $||L^{\leq n}|| \leq d(n)$ for all large n. For any language L, define its *complementary language*, denoted by $\overline{L}$, to be $\Sigma^* - L$.

For a function $t(n) : N \to N$, DTIME($t(n)$) (NTIME($t(n)$)) is the class of languages accepted by (non)-deterministic Turing machines in time $t(n)$. P (NP) is the class of languages accepted by (non-)deterministic polynomial-time Turing machines. E (NE) is the class of languages accepted by (non-)deterministic Turing machines in $2^{O(n)}$ time. EXP (NEXP) is the class of languages accepted by (non-)deterministic Turing machines in time $2^{n^{O(1)}}$. TALLY is the class of languages contained in 1^* and SPARSE is the class of languages in $\bigcup_{c=1}^{\infty}$ Density(n^c). Clearly, TALLY is a subclass of SPARSE. We use P-Sel to denote the class of p-selective sets (Selman 1979). For any language L and function $h : N \mapsto N$, let $L/h = \{x : \langle x, h(|x|)\rangle \in L\}$. For any class $\mathcal{C}$ of languages, co$\mathcal{C}$ is the class of languages L such that $\overline{L} \in \mathcal{C}$ and $\mathcal{C}/h$ is the class of languages L such that $L = L'/h$ for some $L' \in \mathcal{C}$.

For two languages A and B, define the following reductions: (1) A is *polynomial-time many-one reducible* to B, $A \leq_m^p B$, if there exists a polynomial-time computable function $f : \Sigma^* \mapsto \Sigma^*$ such that for every $x \in \Sigma^*$, $x \in A$ if and only if

$f(x) \in B$. (2) A is *polynomial-time truth-table reducible* to B, $A \le_{tt}^{p} B$, if there exists a polynomial-time computable function $f : \Sigma^* \mapsto \Sigma^*$ such that for every $x \in \Sigma^*$, $f(x) = \langle y_1, y_2, \ldots, y_m, T \rangle$, where $y_i \in \Sigma^*$ and T is the encoding of a circuit, and $x \in A$ if and only if $T(B(y_1)B(y_2)\cdots B(y_m)) = 1$. (3) A is *polynomial-time Turing reducible* to B, $A \le_T^P B$, if there exists a polynomial-time oracle Turing machine M such that M^B accepts A. (4) A is *exponential-time Turing reducible* to B, $A \le_T^{\mathrm{EXP}} B$, if there exists an exponential-time oracle Turing machine M such that M^B accepts A. (5) We say $A \le_1^P B$ if $A \le_m^P B$ via a reduction f that is one-to-one.

For a nondeterministic Turing machine M, denote $M(x)[y]$ to be the computation of M with input x on a path y. If $M(x)$ is an oracle Turing machine, $M^A(x)[y]$ is the computation of M with input x on a path y with oracle A.

For two languages A and B, define the following nondeterministic reductions: (1) A is *nondeterministically polynomial-time many-one reducible* to B, $A \le_m^{\mathrm{NP}} B$, if there exists a polynomial-time nondeterministic Turing machine M and a polynomial $p(n)$ such that for every x, $x \in A$ if and only if there exists a path y of length $p(|x|)$ with $M(x)[y] \in B$. (2) A is *nondeterministically polynomial-time truth-table reducible* to B, $A \le_{tt}^{\mathrm{NP}} B$, if there exists a polynomial-time nondeterministic Turing machine M and a polynomial $p(n)$ such that for every $x \in \Sigma^*$, $x \in A$ if and only if there is at least one $y \in \Sigma^{p(|x|)}$ such that $M(x)[y] = (z_1, \ldots, z_m, T)$, where $z_i \in \Sigma^*$, T is the encoding of a circuit, and $T(B(z_1), \ldots, B(z_m)) = 1$. (3) A is *nondeterministically polynomial-time Turing reducible* to B, $A \le_T^{\mathrm{NP}} B$, if there exists a polynomial-time nondeterministic oracle Turing machine M and a polynomial p such that for every $x \in \Sigma^*$, $x \in A$ if and only if there is at least one $y \in \Sigma^{p(|x|)}$ such that $M^B(x)[y]$ accepts. (4) A is *strongly nondeterministically polynomial-time many-one reducible* to B, $A \le_m^{SN} B$, if there exists a polynomial-time nondeterministic Turing machine $M()$ such that $x \in A$ if and only if (1) $M(x)[y] \in B$ for all y that $M(x)[y]$ is not empty; (2) $M(x)[y]$ is not empty for at least one $y \in \Sigma^{n^{O(1)}}$.

For a function $g(n) : N \to N$, we use $A \le_{g(n)-tt}^{\mathrm{NP}} B$ to denote that $A \le_{tt}^{\mathrm{NP}} B$ via a polynomial-time computable function f such that for every $x \in \Sigma^n$, $f(x, y) = (z_1, \ldots, z_m, T)$ and $m \le g(n)$. We use $A \le_{btt}^{\mathrm{NP}} B$ to denote that $A \le_{c-tt}^{\mathrm{NP}} B$ for some constant $c > 0$. For $t \in \{\mathrm{P}, \mathrm{NP}, \mathrm{EXP}\}$, we use $A \le_{g(n)-T}^{t}$ to denote that $A \le_T^t$ via a Turing machine M that makes at most $g(n)$ queries on inputs of length n.

For a class $\mathcal{C}$ of languages, we use $R_r^t(\mathcal{C})$ ($R_{g(n)-r}^t(\mathcal{C})$) to denote the reduction closure of $\mathcal{C}$ under the reduction $\le_r^t$ ($\le_{g(n)-r}^t$), where $r \in \{\mathrm{P}, \mathrm{NP}, SN, \mathrm{EXP}\}$ and $r \in \{m, tt, T\}$. We also use conventional notations for common reduction closures such as $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}_T(\mathrm{NP}) = R_T^{\mathrm{P}}(\mathrm{NP})$ and $\mathrm{EXP}_{n^k-T}^{\mathrm{NP}} = \mathrm{EXP}_{n^k-T}(\mathrm{NP}) = R_{n^k-T}^{\mathrm{EXP}}(\mathrm{NP})$. For a function $l : N \mapsto N$ and a reduction closure R, we use $R[l(n)]$ to denote the same reduction closure as R except that the reductions make queries of length at most $l(n)$ on inputs of length n.

We fix a universal Turing machine $\mathcal{U}$ and define $C(f(n), t(n))$ to be the class of strings x that can be generated by $\mathcal{U}$ from a string of length $c(f(|x|))$ in time $t(|x|)$ for some constant c.

A function $f(n)$ from N to N is time constructible if there exists a Turing machine M such that $M(n)$ outputs $f(n)$ in $f(n)$ steps.

3 Separating NE from $R^{\mathrm{NP}}_{n^{o(1)}-T}$(TALLY)

In this section, we present the main result that NE cannot be reduced to TALLY via polynomial time Turing reduction with the number of queries bounded by $n^{1/\alpha(n)}$ for some polynomial time computable nondecreasing function $\alpha(n)$ (for example, $\alpha(n) = \log\log n$). The proof is a combination of the translational method and the point of view from Kolmogorov complexity.

Lemma 1 *Assume that function $g(n) : N \to N$ is nondecreasing unbounded and function $2^{n^{g(n)}/2}$ is time constructible. Then there exists a language $L_0 \in$* DTIME$(2^{n^{g(n)}})$ *such that $||L_0^n|| = 1$, and for every Turing machine M, M cannot generate any sequence in L_0^n with any input of length $n - \log n$ in $2^{n^{O(1)}}$ time for all large n.*

Proof We use the diagonal method to construct the language L_0. Let $M_1, \ldots, M_k, \ldots$ be an enumeration of all Turing transducers.

Construction:
Input n,
Simulate each machine $M_i(y)$ in $2^{n^{g(n)/2}}$ steps for $i = 1, \ldots, \log n$ and all y of length $n - \log n$.
Find a string x of length n such that x cannot be generated by any machine among $M_1, \ldots, M_{\log n}$ with any input of length at most $n - \log n$.
Put x into L_0.
End of Construction

There are at most $2^{n-\log n+1}$ strings of length at most $n - \log n$. Those $\log n$ machines can generate at most $2^{n-\log n+1} \log n < 2^n$ strings. Since generating each string takes $2^{n^{g(n)/2}}$ steps. This takes $2^n \cdot 2^{n^{g(n)/2}} < 2^{n^{g(n)}}$ time for all large n. □

Theorem 2 *Assume that $t(n)$ and $f(n)$ are time constructible nondecreasing functions from N to N such that* (1) *$t(f(n))$ is $\Omega(2^{n^{g(n)}})$ for some nondecreasing unbounded function $g(n)$, and* (2) *for any constant $c > 0$, $f(n) \le t(n)^{1/c}$ and $f(n) \ge 4n$ for all large n. If $q(n)$ is a nondecreasing function with $q(f(n))(\log f(n)) = o(n)$, then* NTIME$(t(n)) \not\subseteq R^{\mathrm{NP}}_{q(n)-T}$(TALLY).

Proof We apply a translational method to obtain such a separation. We prove by contradiction and assume that NTIME$(t(n)) \subseteq R^{\mathrm{NP}}_{q(n)-T}$(TALLY). Without loss of generality, we assume that $q(n) \ge 1$.

Let L be an arbitrary language in DTIME$(t(f(n))$. Define $L_1 = \{x10^{f(|x|)-|x|-1} : x \in L\}$. It is easy to see that L_1 is in DTIME$(t(n))$ since L is in DTIME$(t(f(n)))$.

By our hypothesis, there exists a set $A_1 \in$ TALLY such that $L_1 \le^{\mathrm{NP}}_{q(n)-T} A_1$ via some polynomial time nondeterministic oracle Turing machine M_1, which runs in polynomial n^{c_1} time for all large n.

Let $L_2 = \{(x, (e_1, \ldots, e_m, a_1 \cdots a_m))$: there is a path y such that $M_1^{A_1}(x10^{f(|x|)-|x|-1})[y]$ accepts and queries $1^{e_1}, \ldots, 1^{e_m}$ in path y and receives answers $a_1 = A_1[1^{e_1}], \ldots, a_m = A_1[1^{e_m}]$ respectively }. Since M_1 runs in time n^{c_1} and $f(n) = t(n)^{o(1)}$, we have L_2 is in NTIME$(f(n)^{c_1}) \subseteq$ NTIME$(t(n))$.

By our hypothesis, there exists a set $A_2 \in$ TALLY such that $L_2 \le^{\mathrm{NP}}_{q(n)-T} A_2$ via some polynomial time nondeterministic oracle Turing machine $M_2()$.

Therefore, for every string x, in order to generate $x \in L$, we need to provide $(e_1, \ldots, e_m, a_1 \cdots a_m)$ and $(z_1, \ldots, z_t, b_1 \cdots b_t)$ such that there exists an accepting path y_1 that $M_1^{A_1}(x10^{f(|x|)-|x|-1})[y_1]$ queries $1^{e_1}, \ldots, 1^{e_m}$ with answers $a_i = A_1(1^{e_i})$ for $i = 1, \ldots, m$ and there exists an accepting path y_2 that $M_2^{A_2}(x, (e_1, \ldots, e_m, a_1 \cdots a_m))[y_2]$ queries $1^{z_1}, \ldots, 1^{z_t}$ with $b_i = A_2(1^{z_i})$ for $i = 1, \ldots, t$. Let n^{c_2} be the polynomial time bound for M_2. We have the following Turing machine M^*.

```
$M^*()$:
Input: a string of $u$ of length $o(n)$.
If $u$ does not have the format $(e_1, \ldots, e_m, a_1 \cdots a_m)(z_1, \ldots, z_t, b_1 \cdots b_t)$,
then return $\lambda$ (empty string).
Extract $(e_1, \ldots, e_m, a_1 \cdots a_m)$ and $(z_1, \ldots, z_t, b_1 \cdots b_t)$ from $u$.
For each $x$ of length $n$
    Simulate $M_2^{A_2}(x, (e_1, \ldots, e_m, a_1 \cdots a_m))$ with the query help from
    $(z_1, \ldots, z_t, b_1 \cdots b_t)$ (by assuming that $b_i = A_2(1^{z_i})$ for $i = 1, \ldots, t$).
Output $x$ if it accepts.
```

It is easy to see that M^* takes $2^{n^{O(1)}}$ time. There exists an accepting path y_1 such that $M_1^{A_1}(x10^{f(|x|)-|x|-1})[y_1]$ makes at most $q(f(n))$ queries, where $n = |x|$. So, we have $m \le q(f(n))$, $e_i \le f(n)^{c_1}$ and $|e_i| \le c_1(\log f(n))$. Therefore, $(e_1, \ldots, e_m, a_1, \ldots, a_m)$ has length $h \le 2(O(q(f(n)) \log f(n)) + q(f(n))) = O(q(f(n)) \log f(n)) = o(n)$. There exists an accepting path y_2 such that $M_2^{A_1}((x, (e_1, \ldots, e_m, a_1 \cdots a_m))[y_2]$ makes at most $q(n + h)$ queries to $1^{z_1}, \ldots, 1^{z_t}$. The length of $(x, (e_1, \ldots, e_m, a_1 \cdots a_m))$ is at most $2(n + h) \le 4n$. So, $t \le q(4n)$. Therefore, $(z_1, \ldots, z_t, b_1 \cdots b_t)$ has length $q(4n) \log((4n)^{c_2}) = O(q(f(n)) \cdot \log f(n)) = o(n)$. Therefore, the total length of $(e_1, \ldots, e_m, a_1 \cdots a_m)$ and $(z_1, \ldots, z_t, b_1 \cdots b_t)$ is $o(n)$. So, $(e_1, \ldots, e_m, a_1 \cdots a_m)$ and $(z_1, \ldots, z_t, b_1 \cdots b_t)$ can be encoded into a string of length $o(n)$. Let L be the language L_0 in Lemma 1. This contradicts Lemma 1 since a string of length n can be generated by $M^*()$ with the input $(e_1, \ldots, e_m, a_1 \cdots a_m)(z_1, \ldots, z_t, b_1 \cdots b_t)$ of length $o(n)$. □

Corollary 3 NE $\not\subseteq R^{\mathrm{NP}}_{n^{1/\alpha(n)}-T}$(TALLY) *for any polynomial computable nondecreasing unbounded function* $\alpha(n) : N \to N$.

Proof Define $g(n) = \lfloor\sqrt{\alpha(n)}\rfloor$, $f(n) = n^{g(n)}$, $q(n) = n^{\frac{1}{\alpha(n)}}$, and $t(n) = 2^n$. By Theorem 2, we have that NTIME$(t(n)) \not\subseteq R^{\mathrm{NP}}_{q(n)-T}$(TALLY). We have that NE $\not\subseteq R^{\mathrm{NP}}_{n^{1/\alpha(n)}-T}$(TALLY) since $R^{\mathrm{NP}}_{n^{1/\alpha(n)}-T}$(TALLY) is closed under $\le^P_m$ reductions and there exists a NE-$\le^p_m$-hard set in NTIME$(t(n))$. □

It is natural to extend Theorem 2 by replacing TALLY by SPARSE. We feel it is still hard to separate NE from R^{NP}_m(SPARSE). The following theorem shows that we can separate NE from R_m^{SN}(SPARSE). Its proof is another application of the combination of translational method with Kolmogorov complexity point of view.

Theorem 4 *Assume that $t_0(n)$ and $t(n)$ are time constructible nondecreasing functions from N to N such that for any positive constant c, $t_0(n)^c = O(t(n))$ and $t(t_0(n)) > 2^{n^{\alpha(n)}}$ for some nondecreasing unbounded function $\alpha(n)$, and $d(n)$ is a nondecreasing function such that $d((t_0(n))^c) = 2^{n^{o(1)}}$. Then* NTIME$(t(n)) \not\subseteq R_m^{SN}$ (Density$((d(n)))$.

Proof Assume that NTIME$(t(n)) \subseteq R_m^{SN}$(Density$(d(n))$. We will derive a contradiction.

Construction of $L^{=n}$: Let S be the sequence of length $n^{1+\frac{1}{k}}$ in L_0 of Lemma 1 with $g(n) = \alpha(n)$, where $n = m^k$ and $k = 100$. Assume that $S = y_1 y_2 \cdots y_{m^2}$, where each y_i is of length m^{k-1}. Let $L^{=n} = \{y_{i_1} y_{i_2} \cdots y_{i_m} : 1 \le i_1 < i_2 < \cdots < i_m \le m^2\}$. Define block$(x) = \{y_{i_1}, y_{i_2}, \ldots, y_{i_m}\}$ if $x = y_{i_1} y_{i_2} \cdots y_{i_m}$. Clearly, $L^{=n}$ contains $\binom{m^2}{m}$ elements.

Define $L_1 = \{x 10^{t_0(|x|)-|x|-1} : x \in L\}$. It is easy to see that L_1 is in DTIME$(t(n))$ since L is in DTIME$(t(t_0(n)))$.

By our hypothesis, there exists a set $A_1 \in$ Density$(d(n))$ such that $L_1 \le_m^{SN} A_1$ via some polynomial time nondeterministic Turing machine $f()$, which runs in polynomial time n^{c_1}. For a sequence z and integer n, define $H(z, n) = \{x \in L^n : f(x)[y] = z$ for some path $y\}$. Therefore, there are a sequence z such that $||H(z, n)|| \ge \frac{\binom{m^2}{m}}{d((t_0(n))^{c_1})}$.

Let $L_2 = \{(x, y) : |x| = |y|$ and there are paths z_1 and z_2 such that $f(x10^{t_0(|x|)-|x|-1})[z_1] = f(y10^{t_0(|y|)-|y|-1})[z_2]\}$. Since $f()$ runs in polynomial time and $t_0(n)^{c_1} = O(t(n))$, we have $L_2 \in$ NTIME$(t(n))$.

By our hypothesis, there exists a set $A_2 \in$ Density$(d(n))$ with such that $L_2 \le_m^{SN} A_2$ via some a nondeterministic Turing machine $u()$, which runs in a polynomial n^{c_2} time.

Define $L_2(x) = \{x_1 : (x, x_1) \in L_2\}$. There exists $x \in L^{=n}$ such that $||L_2(x)|| \ge \frac{\binom{m^2}{m}}{d((t_0(n))^{c_1})}$.

Define $L_2'(x, x') = \{x_2 : u(x, x')[z'] = u(x, x_2)[z_2]$ for some paths z' for $u(x, x')$ and z_2 for $u(x, x_2)\}$. There exists $x' \in L_2(x)$ such that $L_2'(x, x')$ contains at least $\frac{\binom{m^2}{m}}{d((t_0(n))^{c_1})d((t_0(n))^{c_2})}$ elements. We fix x and x'.

Since $||\text{block}(x) \cup \text{block}(x')|| \le 2m$, those $2m$ strings in block$(x) \cup$ block(x') can generate at most $\binom{2m}{m} < \frac{\binom{m^2}{m}}{d((t_0(n))^{c_1})d((t_0(n))^{c_2})}$ sequences of length n in $L^{=n}$ for all large n. Therefore, there is a string $x_3 \in L^{=n}$ such that $x_3 \in L_2'(x, x')$ and block$(x_3) \not\subseteq$ block$(x) \cup$ block(x').

This makes it possible to compress S. We can encode the strings x, x' and those blocks of S not in x_3. The total time is at most $2^{n^{O(1)}}$ to compress S.

Let $y_{i_1} < y_{i_2} < \cdots < y_{i_{m^2}}$ be the sorted list of $y_1, y_2, \ldots, y_{m^2}$. Let $(i_1, i_2, \ldots, i_{m^2})$ be encoded into a string of length $O(m^2(\log n))$. Define $Y = y_{j_1} y_{j_2} \cdots y_{j_t}$, where $\{y_{j_1}, y_{j_2}, \ldots, y_{j_t}\} = \{y_1, \ldots, y_{m^2}\} - (\text{block}(x) \cup \text{block}(x') \cup \text{block}(x_3))$.

We can encode $(i_1, i_2, \dots, i_{m^2})$ into the format $0a_10a_2 \cdot 0a_u11$. We have sequence $Z = (i_1, i_2, \dots, i_{m^2})xx'Y$ to generate S in $2^{n^{O(1)}}$ time. Since at least one block y_i among $y_1, y_2, \dots, y_{m^2}$ is missed in block$(xx'Y)$, $|y_i| = m^{k-1}$, and $|(i_1, i_2, \dots, i_{m^2})| < m^3$, it is easy to see that $|Z| \le n - (\log n)^2$. This brings a contradiction. □

Corollary 5 NE $\not\subseteq R_m^{SN}$(SPARSE).

Proof Let $t(n) = 2^n$, $t_0(n) = n^{\log n}$, and $d(n) = n^{\log n}$. Apply Theorem 4. □

4 On the differences between NE and NP

In this section we investigate the differences between NE-hard sets and NP sets. We use the following well-known result:

Lemma 6 (Ganesan and Homer 1992) *Let H be $\le_m^p$-hard for* NE *and $A \in$* NE. *Then $A \le_1^p H$.*

Theorem 7 *For every set H and $A \subseteq H$ such that H is $\le_m^p$-hard for* NE *and $A \in$* NP, *there exists another set $A' \subseteq H$ such that $A' \in$* NP *and $A' - A$ is not of subexponential density.*

Proof Fix H and A as in the premise and let $A \in \mathrm{NTIME}(n^c)$ for some constant $c > 1$. Let $\{NP_i\}_i$ be an enumeration of all nondeterministic polynomial-time Turing machines such that the computation NP_i on x can be simulated nondeterministically in time $2^{O((|i|+\log(|x|))^2)}$ (Ganesan and Homer 1992). Define $S = \{\langle i, x, y\rangle \ : x, y \in \Sigma^*$ and NP_i accepts $x\}$. Clearly S belongs to NEXP and therefore S is many-one reducible to H via some polynomial-time computable one-one function f. Suppose f can be computed in time n^d for some constant $d > 1$. By Cook (1973), let $B \in \mathrm{NP} - \mathrm{NTIME}(n^{2cd})$. Suppose $B = L(NP_i)$ for some i. For each $x \in \Sigma^*$, define $T_x = \{z : \exists y(|x| = |y|/2 \le |z|$ and $z = f(\langle i, x, y\rangle)\}$. Let $T = \bigcup_{x \in B} T_x$. Clearly $T \in$ NP. Since f reduces S to H, $T_x \subseteq H$ for all $x \in B$ and therefore $T \subseteq H$. We now establish the following claims:

Claim 1 *For infinitely many $x \in B$, $A \cap T_x = \emptyset$.*

Proof Suppose not. Consider the following machine M:

```
0 On input $x$
1 Guess $y$ with $|y| = 2|x|$;
2 Compute $z = f(\langle i, x, y\rangle)$;
3 Accept $x$ if and only if $|z| \ge |x|$ and $z \in A$.
```

Assume that $x \in B$ and $A \cap T_x \ne \emptyset$. Let $z \in A \cap T_x$ and hence there exists y with $|y|/2 = |x| \le |z|$ and $z = f(\langle i, x, y\rangle)$. Thus, M accepts x if it correctly guesses y in line 1. Now assume $x \notin B$. Then $T_x \subseteq \overline{H}$ and hence $A \cap T_x = \emptyset$. Thus, for any z

computed in line 3, $z \notin A$. So M does not accept x. This shows that M decides B for all but finitely many x. However, the machine M runs in time $O(((2|x|)^d)^c) = O((|x|)^{cd})$ for sufficiently large x, which contradicts that $B \notin \mathrm{NTIME}(n^{2cd})$. □

Claim 2 *For any infinite set R, the set $\bigcup_{x\in R} T_x$ is not in* Density$(f(n))$ *for any sub-exponential function $f : N \to N$.*

Proof Let R be an infinite set and $T' = \bigcup_{x\in R} T_x$. Fix a string x. Since f is a one-one function, $\|\{f(\langle i, x, y\rangle)\}_{|y|=2|x|}\| = 2^{2|x|}$. Since there are only $2^{|x|}$ of strings of length less than $|x|$, it follows that there are at least $2^{2|x|} - 2^{|x|} \geq 2^{|x|}$ many strings in T_x. Note that the strings in T_x have lengths at most $\Theta(|x|^d)$ and hence, $\|(T')^{\leq \Theta((|x|)^d)}\| \geq 2^{|x|}$. Since x is arbitrary, this shows that $\bigcup_{x\in R} T_x$ is not Density$(f(n))$ for any sub-exponential function $f : N \to N$. □

Now Let $A' = A \cup T$. By Claims 1 and 2, A' clearly has all the desired properties. This proves Theorem 7. □

Theorem 7 shows that many-one-hard sets for NE are very different from their NP subsets. Namely they're not even sub-exponentially close to their NP subsets. Next we show a stronger result for many-one-hard sets for coNE. We show that the difference between a many-one-hard set for coNE and any of its NP subset has exponential density.

Theorem 8 *Assume that H is a many-one-hard set for* coNE *and $t(n) : N \to N$ is a sub-exponential function. Then for any $A \subseteq H$ with $A \in$* NTIME$(t(n))$, *there exists another set $A' \subseteq H$ such that $A' \in$* NP, *$A' \cap A = \emptyset$, and A' is exponentially dense.*

Proof Fix H and A as in the premise. By a result of Fu et al. (1992, Corollary 4.2), $H' = \overline{H} \cup A$ is many-one hard for NE. Now let f be a polynomial-time one-one reduction from $0\Sigma^*$ to H' and suppose f is computable in time n^d. Let $A' = \{z : z = f(1x)$ for some x with $|x| \leq 2|z|\}$. Clearly $A' \in$ NP and $A' \subseteq \overline{H'}$. Therefore $A' \subseteq H - A$. It remains to show that A' is exponentially dense. For any $n > 0$, let $F_n = \{f(1x)\}_{|x|=2n}$. Since f is one-one, $\|F_n\| = 2^{2n}$. As there are only 2^n strings of length less than n, it follows that there are at least $2^{2n} - 2^n \geq 2^n$ many strings in F_n belonging to A' for each $n > 0$. Note that the maximal length of a string in F_n is $(2n+1)^d$. This shows that $(A')^{\leq (2n+1)^d} \geq 2^n$ for each $n > 0$ and hence, A' is exponentially dense. □

Corollary 9 *Assume that H is a $\leq_m^P$-hard set for* coNE. *Then for $A \subseteq H$ with $A \in$* NP, *there exists another subset $A' \subseteq H$ such that $A' \in$* NP, *$A' \cap A = \emptyset$, and A' is exponentially dense.*

5 Separating NEXP from $\mathrm{P}^{\mathrm{NP}}_{n^k-T}$ for nonuniform reductions

In this section we generalize Mocas's result (Mocas 1996) that NEXP $\not\subseteq \mathrm{P}_{n^c-T}(\mathrm{NP})$ for any constant $c > 0$ to non-uniform Turing reductions.

Lemma 10 *For any positive constants* $k, k' > 0$, $\mathrm{EXP}^{\mathrm{NP}}_{n^k-T} \not\subseteq \mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k'}$.

Proof Burtschick and Lindner (1997) showed that $\mathrm{DTIME}(2^{4f(n)}) \not\subseteq \mathrm{DTIME}(2^{f(n)})/f(n)$ for any function $f\colon N \to N$ with $n \le f(n) < 2^n$. Applying their result with $f(n) = n^{k'}$ yields $\mathrm{EXP} \not\subseteq \mathrm{P}/n^{k'}$ for any $k' > 0$. The lemma follows by noting the fact that Burtschick and Linder's result also holds relative to any oracle. □

Theorem 11 *For any positive constants* $k, k' > 0$, $\mathrm{NEXP} \not\subseteq \mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k'}$.

Proof Assume that $\mathrm{NEXP} \subseteq \mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k'}$ for some $k, k' > 0$. Since $\mathrm{EXP}^{\mathrm{NP}}_{n^k-T} \subseteq \mathrm{P}^{\mathrm{NEXP}}_{T}[n^{k+1}]$ (Mocas 1996), we have $\mathrm{EXP}^{\mathrm{NP}}_{n^k-T} \subseteq \mathrm{P}_T(\mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k'})[n^{k+1}] \subseteq \mathrm{P}^{\mathrm{NP}}_{T}/(n^{k+1})^{k'} \subseteq \mathrm{NEXP}/n^{(k+1)k'} \subseteq (\mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k'})/n^{(k+1)k'} \subseteq \mathrm{P}^{\mathrm{NP}}_{n^k-T}/n^{k''}$ for some $k'' > 0$. The last inclusion is a contradiction to Lemma 10. □

Lemma 12 *For any* $k > 0$, $\mathrm{P}_{n^k-T}(\mathrm{NP} \oplus \textit{P-Sel}) \subseteq \mathrm{P}_{n^k-T}(\mathrm{NP})/n^k$.

Proof Assume that $L \in P_{n^k-T}(NP \oplus \text{P-Sel})$ via polynomial time Turing reduction D. Let A be a P-selective set with order $\preceq$ such that A is an initial segment with $\preceq$ and $L \in P_{n^k-T}(SAT \oplus A)$ via D. Let y be the largest element in A (with the order $\preceq$) queried by $D^{SAT \oplus A}$ among all inputs of length $\le n$. It is easy to see that y can be generated by simulating D with advice of length n^k. When we compute $D^{SAT\oplus A}(x)$, we handle the queries to A by comparing with y. □

By Theorem 11 and Lemma 12, we have the following theorem.

Theorem 13 *For any constant* $k > 0$, $\mathrm{NEXP} \not\subseteq \mathrm{P}_{n^k-T}(\mathrm{NP} \oplus \textit{P-Sel})$.

6 Separating NE from P_{btt}(NP ⊕ SPARSE)

In this section, we separate NE from $P_{btt}(\mathrm{NP} \oplus \mathrm{SPARSE})$ by applying translational method and Kolmogorov complexity again.

Theorem 14 $\mathrm{NE} \not\subseteq P_{btt}(\mathrm{NP} \oplus \mathrm{Density}(n^{\log n}))$.

Proof We prove by contradiction by assuming that $\mathrm{NE} \not\subseteq P_{btt}(\mathrm{NP} \oplus \mathrm{Density}(n^{\log n}))$. A method from Fu (1995) is used in this proof. Let L be an arbitrary language in $\mathrm{DTIME}(2^{f(n)})$, where $f(n) = n^{\log n}$. Define $L_1 = \{x10^{f(|x|)-|x|-1} : x \in L\}$. It is easy to see that $L_1 \in \mathrm{DTIME}(2^n) \subseteq NE$.

By our assumption, there exists a language $S_1 \in \mathrm{Density}(n^{\log n})$ such that $L_1 \in P_{c_1-tt}(SAT \oplus S_1)$ via some $\le^P_{c_1-tt}$ reduction g_1, where c_1 is a constant.

Define $L_2 = \{(x, i) :$ the i-element of $g_1(x10^{f(|x|)-|x|-1})$ is in $SAT\}$. It is easy to see that L_2 is in NE. Therefore, there exists $S_2 \in \mathrm{Density}(n^{\log n})$ such that $L_2 \in P_{c_2-tt}(SAT \oplus S_2)$ via some $\le^P_{c_2-tt}$ reduction g_2, where c_2 is a constant.

In order to make a decision if $x \in L$, it is converted into some constant number of queries to SAT with length $n^{O(1)}$, and constant queries to S_1 of length $n^{O(\log n)}$ and some queries to S_2 of length $n^{O(1)}$. The conversion is through the above two reductions g_1 and g_2. Each query to SAT can be answered in $2^{n^{O(1)}}$ time by using the exhaustive search. Therefore, making decision if $x \in L$ we need $2^{n^{O(1)}}$ time and constant number c_3 queries to $S_1 \oplus S_2$ with length at most $n^{O(\log n)}$. There are $n^{(\log n)^{O(1)}}$ elements of length $n^{O(\log n)}$ in $S_1 \oplus S_2$. Assume that $M^{S_1 \oplus S_2}$ is such a oracle Turing machine for accepting language L.

We derive a contradiction via the Kolmogorov complexity point of view. Let $m = n^{\frac{1}{k}}$ with $k = 100$. Assume that $S_n = y_1 y_2 \cdots y_{m^2}$, where each y_i is of length m^{k-1}. Let $L^{=n} = \{y_{i_1} y_{i_2} \cdots y_{i_m} : 1 \le i_1 < i_2 < \cdots < i_m \le m^2\}$. Assume that S_n cannot be compressed in time $2^{n^{(\log n)/2}}$. The sequence S_n can be found via an exhaustive method to try all strings of length $m^{k-1+2} = m^{k+1} = n^{1+\frac{1}{k}}$ with a universal Turing machine to test if it is compressible. Searching such a S_n can be done in $2^{n^{\log n}}$ time. Therefore, L is in DTIME$(2^{f(n)})$. Define block$(x) = \{y_{i_1}, y_{i_2}, \ldots, y_{i_m}\}$ if $x = y_{i_1} y_{i_2} \cdots y_{i_m}$. Clearly, $L^{=n}$ contains $\binom{m^2}{m}$ elements.

Partition strings in $\Sigma^{\le n^{O(\log n)}}$ into $U_1, \ldots, U_u$ such that 1) each U_i contain a series strings in consecutive lexicographic order in $\Sigma^{\le n^{O(\log n)}}$; and 2)$(S_1 \oplus S_2)(x) = (S_1 \oplus S_2)(y)$ for any strings x and y from the same U_i, where $(S_1 \oplus S_2)(x) = 1$ if $x \in S_1 \oplus S_2$, and $(S_1 \oplus S_2)(x) = 0$ otherwise. Since there are $n^{(\log n)^{O(1)}}$ elements of length $n^{O(\log n)}$ in $S_1 \oplus S_2$, we have $u = n^{(\log n)^{O(1)}}$.

For two strings x and x' in Σ^n, they have the same type if both strings z_i and z_i' are in the same U_j for some j, where z_i and z_i' are the i-th queries made by $M^{S_1 \oplus S_2}(x)$ and $M^{S_1 \oplus S_2}(x')$, respectively. Since M makes at most constant c_3 queries, the total number of types is $n^{(\log n)^{O(1)}}$.

There exists one type $T \subseteq L^{=n}$ that contains $\frac{||L^{=n}||}{n^{(\log n)^{O(1)}}} = \frac{\binom{m^2}{m}}{n^{(\log n)^{O(1)}}} = \Omega(\binom{m^{2-o(1)}}{m})$ strings in $L^{=n}$. Assume that $U_{i_1}, \ldots, U_{i_{c_3}}$ are the sets that holds all queries generated by strings from T.

There are many strings in $L^{=n}$ have the queries to $S_1 \oplus S_2$ in the same type. For each U_{i_j}, let x_j be the string in T such that it generates the least queries in U_{i_j} under lexicographic order, and let x_j' be the string in T such that it generates the largest queries in U_{i_j} under lexicographic order.

Using $\{x_1, x_1', \ldots, x_{c_3}, x_{c_3}'\}$, we can generate all strings in T in $2^{n^{O(1)}}$ time by checking the range for all queries to $S_1 \oplus S_2$. This can generate $\Omega(\binom{m^{2-o(1)}}{m})$ many strings that contain $m^{2-o(1)}$ blocks. Note that the number of blocks of the strings in $\{x_1, x_1', \ldots, x_{c_3}, x_{c_3}'\}$ is only $O(m)$. Assume the blocks for the strings in T are $y_{i_1}, \ldots, y_{i_t}$. With information $\{x_1, x_1', \ldots, x_{c_3}, x_{c_3}'\}$, $i_1, \ldots, i_t$, and the blocks $\{y_1, y_2, \ldots, y_{m^2}\} - \{y_{i_1}, \ldots, y_{i_t}\}$ (the sets are ordered), we can recover S_n in $2^{n^{O(1)}}$ time. This makes S_n compressible in $2^{n^{(\log n)/2}}$ time. A contradiction is brought. □

Corollary 15 NE $\not\subseteq P_{btt}$(NP $\oplus$ SPARSE).

7 Conclusions

We derived some separations between NE and other nondeterministic complexity classes. The further research along this line may be in separating NE from P_T^{NP}, and NE from BPP, which is a subclass of P/Poly.

Acknowledgements We thank unknown referees for their helpful comments. Bin Fu is supported in part by National Science Foundation Early Career Award 0845376.

References

Berman L, Hartmanis J (1977) On isomorphisms and density of NP and other complete sets. SIAM J Comput 6(2):305–322

Buhrman H, Torenvliet L (1994) On the cutting edge of relativization: the resource bounded injury method. In: ICALP 1994. Lecture notes in computer science, vol 820. Springer, Berlin, pp 263–273

Burtschick H-J, Lindner W (1997) On sets Turing reducible to p-selective sets. Theory Comput Syst 30:135–143

Cai J, Sivakumar D (1999) Sparse hard sets for P: resolution of a conjecture of Hartmanis. J Comput Syst Sci 58(2):280–296

Cook S (1973) A hierarchy for nondeterministic time complexity. J Comput Syst Sci 7(4):343–353

Fu B (1993) On lower bounds of the closeness between complexity classes. Math Syst Theory 26(2):187–202

Fu B (1995) With quasilinear queries EXP is not polynomial time Turing reducible to sparse sets. SIAM J Comput 24(5):1082–1090

Fu B, Li H, Zhong Y (1992) Some properties of exponential time complexity classes. In: Proceedings 7th IEEE annual conference on structure in complexity theory, pp 50–57

Ganesan K, Homer S (1992) Complete problems and strong polynomial reducibilities. SIAM J Comput 21(4):733–742

Hemaspaandra L, Ogihara M (2002) The complexity theory companion. Texts in theoretical computer science—an EATCS series. Springer, Berlin

Hemaspaandra L, Torenvliet L (2003) Theory of semi-feasible algorithms. Springer, Berlin

Homer S, Selman A (2001) Computability and complexity theory. Texts in computer science. Springer, New York

Karp R, Lipton R (1980) Some connections between nonuniform and uniform complexity classes. In: Proceedings of the twelfth annual ACM symposium on theory of computing, pp 302–309

Mahaney S (1982) Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. J Comput Syst Sci 25(2):130–143

Mocas S (1996) Separating classes in the exponential-time hierarchy from classes in PH. Theor Comput Sci 158:221–231

Ogihara M, Tantau T (2004) On the reducibility of sets inside NP to sets with low information content. J Comput Syst Sci 69:499–524

Ogiwara M (1991) On P-closeness of polynomial-time hard sets. Unpublished manuscript

Ogiwara M, Watanabe O (1991) On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. SIAM J Comput 20(3):471–483

Selman A (1979) P-selective sets, tally languages and the behavior of polynomial time reducibilities on NP. Math Syst Theory 13:55–65

Yesha Y (1983) On certain polynomial-time truth-table reducibilities of complete sets to sparse sets. SIAM J Comput 12(3):411–425