A NOTE ON NON-COMPLETE PROBLEMS IN $\mathrm{NP}_{\mathbb{R}}$

S. BEN-DAVID, K. MEER, AND C. MICHAUX

ABSTRACT. This note deals with the structure of the class $\mathrm{NP}_{\mathbb{R}}$ introduced by Blum, Shub and Smale [5]. It is shown that, assuming $\mathrm{NP}_{\mathbb{R}} \not\subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{poly}$, there exists a problem in $\mathrm{NP}_{\mathbb{R}} \setminus (\mathrm{P}_{\mathbb{R}}/\mathrm{poly})$ which is not $\mathrm{NP}_{\mathbb{R}}$-complete (w.r.t $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ reductions). It also clarifies the scope of a padding technique used in a former similar result by Ladner [11] concerning the structure of the class NP (in the common, Turing machine, model).

1. Introduction

Throughout this paper we assume the reader to be aware of the computational model of Blum, Shub and Smale (shortly BSS) over $\mathbb{R}$ and $\mathbb{C}$ as introduced in [5], see also [4].

Ladner [11] has shown the existence of non-complete problems in $NP \setminus P$ (in the classical Turing machine setting) if $P \neq NP$ is assumed. In [12] this result was extended to the computational model of Blum, Shub and Smale over algebraically closed fields. The main ingredient of these results is a diagonalization argument on a countable set of machines (referred to in the sequel as Ladner's padding technique). In the classical case the set of all machines clearly is countable. This is not true for BSS-machines over uncountable structures like $\mathbb{R}$ or $\mathbb{C}$. In the case of complex numbers this difficulty is overcome in [12] by using a transfer principle proved by Blum, Cucker, Shub and Smale [3]; it allows to work over a countable set of machines again.

Here we consider the BSS-model of computation over $\mathbb{R}$. The above mentioned transfer principle is not known to hold true in this framework. However by taking advantage of a careful analysis of Ladner's padding technique (no matter which variant – [11] or[12] – we choose) as well as of the quantifier elimination property of the reals (Tarski's Theorem) we are able to prove nonuniform analogues of Ladner's and Malajovich-Meer's results ([11], [12]), in the real case. Moreover it is clear from the proofs that the results proved here are valid in the broader context described in [10] and [14]. Together with the fact that the particular subclass of *P/poly* which is used in our proof of Theorem 1 is equal to the class P in recursively saturated structures (see [13] and remarks below), this turns out in particular to reprove the results in [11] and [12].

2. The scope of Ladner's argument

In this section we give a brief sketch of Ladner's original argument. We hope this will help the reader to be convinced of the validity of the arguments performed in the main section.

As we said in the introduction, Ladner's argument is mainly a diagonalization trick on a countable set of machines. Note that in today's terminology it is a consequence of Schöning's general uniform diagonalization theorem, see [16]. More precisely it works as follows :

First one notes that the families $\mathfrak{P}$ of all polynomial time machines on the one hand side and $\mathfrak{R}$ of all polynomial time oracle machines on the other are effectively enumerable. Taking such an effective enumeration of these machine classes and assuming $P \neq NP$ a problem of intermediate complexity is constructed as follows: start with a NP-complete problem and change it on specific input sizes to coincide with an easy (i.e. polynomial time solvable) problem. This is done in such a way that step by step alternately both the machines from $\mathfrak{P}$ and $\mathfrak{R}$ fail: keeping the problem difficult on sufficiently many input sizes the machines from $\mathfrak{P}$ will believe it to be a difficult one; making it easy on sufficiently many other input sizes the machines from $\mathfrak{R}$ will believe it to be too easy to use it as an oracle for solving another NP-complete problem.

The main difficulty here is to guarantee the effective computation of the "failure dimensions". Let E be an effective enumeration of all polynomial time machines and let L be a problem in $NP \setminus P$. Thus for every $i \in \mathbb{N}$ the polynomial time machine $E(i)$ fails to decide L on some input. Now a failure dimension function $n : \mathbb{N} \to \mathbb{N}$ (depending on L) is a computable function such that for any $j \leq i$ the machine $E(j)$ fails to decide L on some input of size at most $n(i)$. Since the class $\mathfrak{P}$ is recursively presentable in the Turing setting this function n is recursive (in the Turing model [1])

It is obvious that this proof heavily relies on the countability of the underlying structure; the machines from $\mathfrak{P}$ and $\mathfrak{R}$ must be enumerated in order to be fooled one by one. Therefore at a first sight it is not clear what is behind the technique to allow a transformation to uncountable settings like BSS-machines over $\mathbb{R}$ or $\mathbb{C}$.

As mentioned in the introduction over $\mathbb{C}$ one can reduce the problem once again to a discrete one by means of a transfer principle given in [4], see [12]. For real closed fields such a principle is not known to hold true. Thus we are forced to proceed in a different manner.

In a first step we will separate BSS computations over $\mathbb{R}$ into a discrete algorithmic part and a continuous part describing the use of real machine constants.

[1]As shown in [16] these arguments have little to do with P and NP, but with recursive presentable classes. Meanwhile in the rough discussion below we will keep the symbols P and NP as a generic notation for two classes of complexity such that $NP \not\subseteq P$ and NP admits complete problems with respect to P reductions.

Note that such a distinction is commonly used in the BSS framework (for example it is the starting point of developing a descriptive complexity theory over $\mathbb{R}$, see [9]). The main idea of the present paper then is to apply quantifier elimination in order to establish recursiveness of the failure dimension function in the BSS setting.

Let us remark that the decomposition of machines causes our results to be nonuniform over $\mathbb{R}$. However, if we analyze the main theorem over ω-saturated structures like $\mathbb{C}$ or $\{0,1\}$ our proof can be turned into a uniform one. Thus, the main theorem reproves Ladner's original one as well as Malajovich-Meer's result over $\mathbb{C}$ - this time by means of quantifier elimination.

We start with a basic definition.

Definition 2.1. a) *Let $\mathfrak{P}$ be the (uncountable) set of polynomial time BSS machines (either real or complex). We say that $\mathfrak{P}$ admits a countable covering $E^\sim$* iff it can be decomposed into subsets $E^\sim(i), i \in \mathbb{N}$:

$$\mathfrak{P} = \bigcup_{i\in\mathbb{N}} E^\sim(i)$$

b) $E^\sim$ is called *effective covering* iff one can enumerate some effective representation of the family of subsets $E^\sim(i), i \in \mathbb{N}$.

c) Let L be a problem not decidable by any machine in $\mathfrak{P}$. A *failure dimension function* (depending on L) is a function $n^\sim : \mathbb{N} \to \mathbb{N}$ such that the following holds: $\forall i \in \mathbb{N}$ and for every machine M in the subset $E^\sim(i)$ there is an input x of size less than $n^\sim(i)$ such that M fails to decide L on x.

After showing the existence of a recursive failure dimension function for $\mathfrak{P}$ we will perform Ladner's padding technique on the family of subset $E^\sim(i)$ of $\mathfrak{P}$.

Let us remark that computability of function $n^\sim$ says more than L is not in complexity class P. Consider a machine M which works (nonuniformly) as follows: associated to M is one fixed index $i \in \mathbb{N}$. Now for every $n \in \mathbb{N}$ there is a machine M_n in $E^\sim(i)$ such that for all inputs of dimension n machine M simulates M_n. Denote by $P/E^\sim$ the resulting complexity class of problems decided by such a machine M. Now if the conditions of definition 2.1 are satisfied the padding technique will give us the desired problem $\tilde{L}$ in $NP \setminus P/E^\sim$ which is not NP-complete (and not only $\tilde{L} \in NP \setminus P$). Moreover, we can even allow non-uniform reductions.

In the next section we exhibit two cases concerning the BSS-model of computation where we apply these ideas by proving that the conditions of definition 2.1 are satisfied.

3. The main results

Let us begin with defining a real version of $P/poly$ as well as a subclass of it. We suppose the reader to be familiar with the notion of an algebraic circuit, see for example [4] or [14].

Definition 3.1. *A problem L is in class* $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ *if and only if there exists a family $\{C_n\}_{n\in\mathbb{N}}$ of algebraic circuits of polynomial size in n such that C_n decides $L\cap\mathbb{R}^n$ for all $n\in\mathbb{N}$. We will call* $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ *machines such families of algebraic circuits. Let us remark that in the BSS-setting over $\mathbb{R}$ such circuits can involve real constants.*

Definition 3.2. *A basic machine over $\mathbb{R}$ in the BSS-setting denotes a BSS-machine M without constants with two blocks of input variables, one block $\bar{x}$ taking values in $\mathbb{R}^\infty$, the other one $\bar{y}$ in $\mathbb{R}^l$ ($l<\infty$, l is a constant of M). Here, as in* [5]*, $\mathbb{R}^\infty$ denotes the set of all finite sequences of real numbers.*

The set of basic machines is clearly countable. Now for any substitution $\bar{c}$ for $\bar{y}$ we have a BSS-machine $M(\bar{c})$ over $\mathbb{R}$ (with constants $\bar{c}\in\mathbb{R}^l$) and conversely any BSS-machine can be obtained by (uniform) substitution from a basic machine. We summarize these facts by saying that any BSS-machine has an underlying basic machine.
Let us also remark that this definition extends in an obvious way to nonuniform machines, so we will freely speak of basic $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ machines (but there are uncountably many ones).

Beside the above introduced general notion of nonuniformity there is a weaker one. It was investigated in [13] and turns out to be important especially when applying our results to ω-saturated structures.

Definition 3.3. [13] *A problem L is in class* $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *if and only if there exists a polynomial time basic machine M and for every $n\in\mathbb{N}$ a tuple $\bar{c}_n$ of real constants for M such that $M(\bar{c}_n)$ decides $L\cap\mathbb{R}^n$ upto size n.*

Remark 3.4. *Using the transformation from BSS programs to algebraic circuits given in* [8] *we easily see the relations* $\mathrm{P}_{\mathbb{R}}\subset\mathrm{P}_{\mathbb{R}}/\mathrm{const}\subset\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ *to hold true.*

Theorem 3.5. *Assume* $\mathrm{NP}_{\mathbb{R}}\not\subseteq\mathrm{P}_{\mathbb{R}}/\mathrm{const}$. *Then there exists a problem $\tilde{L}$ in* $\mathrm{NP}_{\mathbb{R}}\setminus(\mathrm{P}_{\mathbb{R}}/\mathrm{const})$ *which is not* $\mathrm{NP}_{\mathbb{R}}-$ *complete with respect to* $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *- reductions.*

Proof. We have to show that the conditions of definition 2.1 are satisfied. We start with an enumeration $M_1,M_2,\dots$ of polynomial time basic machines - just by performing a common enumeration as in [1] on the set of basic machines. This enumeration provides us with an effective covering $E^{\sim}$of the set of polynomial time machines : $E^{\sim}(i)$ can be defined as the set of those polynomial time machines for which the underlying basic machine is one of the M_j's, for $j\le i$. So, for each M_j there is a corresponding polynomial time bound p_j. Let p_j and the machine M_j be fixed. Let L be a problem in $\mathrm{NP}_{\mathbb{R}} \setminus (\mathrm{P}_{\mathbb{R}}/\mathrm{const})$; for any choice $\bar{c}$ of the constants of M_j, the resulting machine $M_j(\bar{c})$ fails to decide L on some input. We claim that this statement can be strengthened to the existence of a fixed n_j such that, for every $\bar{c}$, $M_j(\bar{c})$ fails to decide L on some input of dimension $\leq n_j$. Otherwise we could find for each dimension n, a tuple $\bar{c}_n$ of constants such that the nonuniform machine $M_j(\bar{c}_n)$ would solve L on all inputs of dimension $\leq n$ contradicting the assumption $L \notin \mathrm{P}_{\mathbb{R}}/\mathrm{const}$.

Let us now show how this value of n_j may be computed by means of effective quantifier elimination. This is the crucial part of the proof. Indeed, for every dimension n we can effectively write down a first-order formula $\Phi_n(\bar{x}, \bar{y})$, $\bar{x} = x_1, \ldots, x_n$, in the natural language of ordered rings $\mathfrak{L}$, which expresses the behavior of the machine $M_j(\bar{y})$ for inputs of dimension n (for all details see [13]). Applying the assumption that L is an $\mathrm{NP}_{\mathbb{R}}$ problem, we find, for every dimension n, a $\mathfrak{L}$–first-order formula $\rho_n(\bar{x})$ defining the restriction of L to the set of n - dimensional inputs (see [5] or [13]). For increasing n we now check the truth of the sentence

$$\Theta_n \;\equiv\; \forall\, \bar{y}\, \exists\, \bar{x} \;\; \neg\; (\Phi_n(\bar{x}, \bar{y}) \;\Leftrightarrow\; \rho_n(\bar{x}))$$

by effectively eliminating quantifiers. "We check effectively" here refers to a BSS-algorithm over the reals. Note that the formulas $\rho_n(\bar{x})$ can involve real parameters.

We are sure to find n such that the sentence Θ_n is true. The first such n is the desired value n_j we are looking for. Moreover, that gives a computable process to determine such a value n_j given a problem $L \in \mathrm{NP}_{\mathbb{R}} \setminus (\mathrm{P}_{\mathbb{R}}/\mathrm{const})$ and a polynomial time basic machine, M_j. It is clear that the function $n^{\sim}$, defined by $n^{\sim}(i)$ as the maximum of the values n_j for $j \leq i$, satisfies the requirement of definition 2.1.

We are now in the scope of Ladner's argument as explained in the previous section. So we can apply Ladner's padding technique for constructing the problem $\tilde{L}$ we are looking for. We start with L as initial problem; it will be changed step by step into an easier one $\tilde{L} \in \mathrm{NP}_{\mathbb{R}}$ such that the following conditions hold true:

i) L is not $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ reducible to $\tilde{L}$
ii) $\tilde{L}$ is not in $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$.

In order to fulfill these requirements we will need to construct a failure dimension function $n^{\sim}$ together with a second function $m^{\sim} : \mathbb{N} \to \mathbb{N}$ (which is a failure dimension function for the class of machines performing $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ reductions) such that

- $n^{\sim}(i) < m^{\sim}(i) < n^{\sim}(i+1)\ \forall i \in \mathbb{N}$;
- for all inputs of dimension j such that there is an $i \in \mathbb{N}$ with $m^{\sim}(i) < j \leq n^{\sim}(i+1)$ we define $\tilde{L}$ to equal L;

- for all inputs of dimension j such that there is an $i \in \mathbb{N}$ with $n^{\sim}(i) < j \leq m^{\sim}(i)$ we define $\tilde{L}$ to be the empty set.

Here the computability of $n^{\sim}$ and $m^{\sim}$ is crucial: it provides $\tilde{L}$ to be computable. Using some well known further arguments $\tilde{L}$ moreover then can be guaranteed as well to belong to $\mathrm{NP}_{\mathbb{R}}$.

Since the corresponding arguments are a step by step transposition of the proofs in [11] or [12] we close with this short description. □

Remark 3.6. *In fact going into the details of the padding technique allows to establish the following result: assume L to be a problem decidable within some time bound $g(n)$ for inputs of dimension n and $L \notin \mathrm{P}_{\mathbb{R}}/\mathrm{const}$. Then there exists a problem $\tilde{L} \notin \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ such that $\tilde{L}$ is $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ reducible to L but not vice versa.*
This is due to the fact that for a decidable problem with time bound $g(n)$ we can again effectively write down a first-order formula Φ_n as above.

The basic ingredient of the proof of Theorem 1 can also be used to obtain the following result:

Theorem 3.7. *Assume $\mathrm{NP}_{\mathbb{R}} \not\subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{poly}$. Then there exists a problem $\tilde{L}$ in $\mathrm{NP}_{\mathbb{R}} \setminus (\mathrm{P}_{\mathbb{R}}/\mathrm{poly})$ which is not $\mathrm{NP}_{\mathbb{R}}-$ complete with respect to reductions in $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$.*

Proof. The main task is to find a good effective covering of the set of $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ machines to mimic the proof of Theorem 1. But here even the set of basic $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ machines is uncountable. Let us consider the list of polynomials $n^k + k$ for $k \in \omega$. Any polynomial is bounded above by some member of the list. Define the sets $E^{\sim}(i)$, for $i \in \omega$ as the set of families $\{C_n\}_{n \in \mathbb{N}}$ of algebraic circuits of size bounded by $n^i + i$ for each n. They provide a covering of the set of $\mathrm{P}_{\mathbb{R}}/\mathrm{poly}$ machines. This covering is effective in the sense that if we limit machines to handle inputs of size less than n, the underlying basic nonuniform machines of machines in $E^{\sim}(i)$ are finitely many (depending on n) since the size of their circuits is bounded by $n^i + i$. This again allows to effectively check the existence of an input dimension $n^{\sim}(i)$ such that every member of class $E^{\sim}(i)$ fails to decide L on some input of dimension $\leq n^{\sim}(i)$. Indeed for each dimension n and for each basic algebraic circuit we can write down a $\mathfrak{L}$–first-order formula (as we claim in the proof of Theorem 1 for basic machines) which expresses the behaviour of the circuit. So as in the proof of Theorem 1 we can write and check (again by means of quantifier elimination) a formula Θ_{n,C_n}, depending on n and the circuit, which is true if and only if the circuit fails to decide L on inputs of size n. We point out that instead of checking one single formula of type Θ_n we have to check finitely many, one for each basic circuit of size less than $n^i + i$ with n inputs. The process of writing down and checking formula will stop since we are sure by our assumption on L ($L \notin \mathrm{P}_{\mathbb{R}}/\mathrm{poly}$) that there exists a n such that any machine in $E^{\sim}(i)$ fails to decide L on inputs of dimension less than n. □

Let us finish with some concluding remarks:

a) The proof can be performed in any first order structure which admits effective quantifier elimination in a finite first order language. In this case Poizat has shown decidability of class NP and existence of complete problems [10] and [14]. Thus our result is valid for any algebraically or real closed field as well as any finite structure.
b) In recursively saturated structures it is easy to see that $P/const = P$ (see [13]). Since any algebraically closed field of infinite transcendence degree as well as finite structures are recursively saturated, our result re-proves those in [11] and [12], the latter without using the transfer principle of [3].
c) Soon after a preliminary version [2] of this work was done, the authors were informed that Poizat obtained results similar to those in this note ([15]). Chapuis and Koiran study the class of complexity $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ in a forthcoming paper [7]. Bürgisser obtained Ladner like diagonalization results for Valiant's class of p-definable polynomials, see [6].

4. Acknowledgements:

All authors were supported by EC Working group NeuroCOLT under contract 8556. The second and third author were partially supported by Wissenschaftsministerium des Landes Nordrhein-Westfalen and the MSRI, Berkeley, where a major revision of an earlier version was done. We want to thank the referees for their careful reading and their useful remarks to improve the presentation.

References

[1] T. Baker, J. Gill, R. Solovay, *Relativizations of the $P = ?NP$ question*, SIAM Journal Computing **4** (1975), 431-442.
[2] S. Ben-David, K. Meer, C. Michaux, *A Note on Non-complete Problems in* $\mathrm{NP}_{\mathbb{R}}$, Esprit Working Group NeuroCOLT, Technical Report NC-TR-1997-007 (1997).
[3] L. Blum, F. Cucker, M. Shub, S. Smale, *Algebraic settings for the problem "$P \neq NP$"*, Proceedings of the 25th AMS-SIAM Summer Seminar in Applied Mathematics, Park City 1995, ed.: J. Renegar, M. Shub, S. Smale, A.M.S Providence (1996), 125-144.
[4] L.Blum, F.Cucker, M. Shub, and S. Smale, *Complexity and Real Computation*, Springer-Verlag, Berlin, (1998).
[5] L. Blum, M. Shub, S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc. **21** (1989), 1-46.
[6] P. Bürgisser, *On the structure of Valiant's complexity classes*, Proc. of the 15th Symposium on Theoretical Aspects of Computer Science STACS '98 (Lecture Notes in Computer Science), Springer (1998), 194-204.
[7] O. Chapuis, P. Koiran, *First-order structures saturated for polynomial time*, to appear in: Annals of Pure and Applied Logic.
[8] F. Cucker, A. Torrecillas, *Two P - complete problems in the theory of the reals*, Journal of Complexity, **8** (1992), 454-466.

[9] E. Grädel, K. Meer, *Descriptive complexity theory over the real numbers*, in: J. Renegar, M. Shub, and S. Smale (editors): The Mathematics of Numerical Analysis, Lectures in Applied Mathematics **32**, American Mathematical Society (1996), 381-404.
[10] J.B. Goode, *Accessible telephone directories*, Jour. of Symb. Logic, **59**, 1 (1994), 92-105.
[11] Richard Ladner, *On the structure of polynomial time reducibility*, Journal of the ACM **22** (1975), 155-171.
[12] G. Malajovich, K. Meer, *On the Structure of* $\mathrm{NP}_{\mathbb{C}}$, SIAM Journal on Computing **28**, No.1 (1999), 27-35.
[13] C. Michaux, *$P \neq NP$ over the nonstandard reals implies $P \neq NP$ over* $\mathbb{R}$, Theoretical Computer Science **133** (1994), 95-104.
[14] B. Poizat, *Les petits cailloux*, Ed. Aléas, Lyon 1995.
[15] B. Poizat, personal communication to the third author and *Gonflette dans les modèles ω-saturés*, exposé au groupe de travail LIP–IGD "Complexité algébrique", Lyon, 1996.
[16] U. Schöning, *A uniform approach to obtain diagonal sets in complexity classes*, Theoretical Computer Science, **18** (1982), 95-103.

S. Ben-David, Computer Science Department, Technion, Haifa 32000, Israel
E-mail address: shai@cs.technion.ac.il

K. Meer, RWTH Aachen, Lehrstuhl C für Mathematik, Templergraben 55, D - 52062 Aachen, Germany
E-mail address: meer@alpha.mathc.rwth-aachen.de

C. Michaux,, Université de Mons-Hainaut, Institut de Mathématique et d'Informatique, Avenue Maistriau 15, B - 7000 Mons Belgium
E-mail address: chrmich@sun1.umh.ac.be