



ELSEVIER

Information Processing Letters 72 (1999) 83–90

Information
Processing
Letters

www.elsevier.com/locate/ipl

On the computational structure of the connected components of a hard problem

Martin Matamala^{a,*}, Klaus Meer^{b,1}

^a *Universidad de Chile, Dpto. de Ingeniería Matemática, Beaucheff 850, Casilla 170-3, Santiago, Chile*

^b *Fakultät für Informatik, TU Chemnitz, D-09107 Chemnitz, Germany*

Received 4 August 1998; received in revised form 18 October 1999

Communicated by P.M.B. Vitányi

Abstract

The study of sparse sets has tremendous importance in Turing complexity theory. Thus it is a natural task to work out a related notion for real number models of computation. This has not fully been done so far. In the present paper we suggest such a notion based on the computational structure of the connected components a set has. Even though our notion of well-structured sets is different in spirit from the classical sparseness property we will show that it shares some important features with the latter. We are going to analyze the (non-)existence of well-structured complete sets within the Blum–Shub–Smale model of computation over the reals with linear operations and equality respectively inequality. Relations to exponential time classes are also drawn. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Connected components; Sparseness; Computational complexity over the reals

1. Introduction

One of the main important theorems on the structure of NP-complete sets is due to Mahaney [14]. It states that no sparse NP-hard sets exist unless $P = NP$. Here sparseness denotes “small” sets in the following sense: let Σ be the finite alphabet $\{0, 1\}$; a set $S \subset \Sigma^*$ is sparse if and only if there exists a polynomial p which bounds for all $n \in \mathbb{N}$ the census function

$$c(n) := |S \cap \Sigma^n|$$

from above.

Beside the classical approach to complexity provided by Turing algorithms on finite alphabets there is an increasing interest in computability and complexity over the reals. This is due to the fact that for many computational problems real numbers come into play quite naturally. Just to mention a few examples consider some of the most important algorithms in “classical” mathematics such as Newton’s method (see [3]) or the investigation of neural networks which involve real (for example, physical) constants in the neurons [13]. An appropriate computational model was introduced in 1989 by Blum et al. [5]. It investigates a notion of uniform algorithms over the real numbers (as well as other structures). Real number complexity classes $P_{\mathbb{R}}$ and $NP_{\mathbb{R}}$ of problems being deterministically decidable respectively non-deterministically verifiable in polynomial time by such an algorithm are

* Corresponding author. Email: mmatamala@dim.uchile.cl. Partially supported by Fondecyt 1970398.

¹ Email: klaus.meer@informatik.tu-chemnitz.de. Partially supported by Fondap on Applied Mathematics.

defined straightforwardly. The question of whether $P_{\mathbb{R}}$ equals $NP_{\mathbb{R}}$ turns out to be of major importance in this framework as well. Moreover, also specific variants of the Blum–Shub–Smale model have been studied. In particular, so called linear models provide (beside being interesting in their own) close relations to open problems in the Turing setting (see [8]). For an introduction into the field confer [4]. A survey on recent results is given in [18].

We suppose the reader to be familiar with the Blum–Shub–Smale (shortly: BSS) approach and adapt the commonly used notations.

In discrete complexity theory, i.e., over $\Sigma = \{0, 1\}$, Mahaney’s theorem states that an NP-hard problem most likely must have many elements.

Because of its importance in classical complexity theory it is a natural task to look for a meaningful notion of sparseness in the BSS model as well. There has not been much work on this topic so far (according to our knowledge only two papers deal with such a real version of sparseness, see [7,17]).

The general idea of sparseness is to reduce the value of a specific measure for sets by an exponential factor. Over finite alphabets this measure is taken to be the number of elements. But what should be a significant counterpart in a real setting? In [7] the authors propose the topological dimension of a set. In general, if S is a decision problem over $\mathbb{R}^{\infty} := \bigcup_{i=1}^{\infty} \mathbb{R}^i$, i.e., $S \subset \mathbb{R}^{\infty}$, the intersections $S \cap \mathbb{R}^n$ can have a topological dimension close to n . Thus in [7] a sparse set is defined to be a set $S \subset \mathbb{R}^{\infty}$ for which these quantities are bounded by $\log(n) \forall n \in \mathbb{N}$. For this notion it is then shown the non-completeness of any sparse set in the additive BSS model with branching on equality. This corresponds to Mahaney’s theorem since in the above setting P is known to be different from NP (see [12,15]). Unfortunately, nothing is known about the computational behavior of sets having a low dimension in more powerful variants of the BSS model, i.e., if branching on inequalities is allowed. Here it is not clear whether this notion of sparse real sets can serve as an analogue to the classical case.

In the present paper we propose to study another property of real number problems which is related to the computational structure of the connected components a set has. This is different in spirit to classical sparseness in that a computational condition enters

into consideration—the reason why we prefer not to speak of sparseness but of well-structured sets. However, as will be shown our approach shares many of the properties of sparse sets in the Turing framework even for ordered BSS models. The latter includes Mahaney like theorems and connections to exponential time classes for linear BSS machines as well as relations with polynomial generators in arbitrary BSS models. Moreover, reduced to the classical framework our notion of being well-structured captures the original sparseness notion.

Let us be a little more precise: Instead of the topological dimension we start with another important topological measure; in general, any semi-algebraic set in \mathbb{R}^n defined by a Boolean combination of polynomial equalities and inequalities such that all involved polynomials have a degree bounded by a fixed number has a bounded number of connected components; the bound is simply exponential in the number n of variables (see, for example, [2,11] and the literature cited there). Moreover, given two points $x, y \in \mathbb{R}^n$ together with such a polynomial system, it is decidable in simply exponential time in n whether x and y belong to the same connected component of the semi-algebraic set defined by the system [11]. Since the n -dimensional part of any decision problem in NP^{Δ} , where Δ stands for the full, ordered linear or unordered linear BSS model (see below), defines a semi-algebraic set we take the above observation as starting point and propose the following notion of well-structured problems over \mathbb{R} : A set $S \subset \mathbb{R}^{\infty}$ is called well-structured iff there is a polynomial p which bounds the number of connected components of $S \cap \mathbb{R}^n \forall n \in \mathbb{N}$ and the problem of checking whether two points belong to the same connected component of $S \cap \mathbb{R}^n$ is semi-decidable in polynomial time (for a precise definition see Section 2). This definition asks for some comments and explanations which will be given in Section 2.

The main result of this paper is an analogue of Mahaney’s theorem in the linear model with order. Some further aspects of our notion of well-structured sets are discussed.

2. Basic notations; well-structured sets

Let \mathbb{R}^{∞} be the disjoint union of all the spaces \mathbb{R}^n , $n \in \mathbb{N}$. For any $x \in \mathbb{R}^n$ let $size(x) := n$. A decision

problem over \mathbb{R} is a subset $T \subset \mathbb{R}^\infty$ and its n -dimensional part T_n is given as

$$T_n := T \cap \mathbb{R}^n.$$

We consider linear restrictions of the BSS model, i.e., we allow as operations addition, subtraction and scalar multiplication by a scalar belonging to a fixed, finite set of machine constants. Referring to order free linear machines (or linear machines branching on equality, respectively) we mean linear machines which can only perform equality branches; similarly the term ordered linear machines (or linear machines branching on inequality, respectively) refers to machines performing branches “ $x \geq 0$?”. The corresponding analogues of complexity classes $\mathbb{P}_{\mathbb{R}}$ and $\mathbb{NP}_{\mathbb{R}}$ are denoted by $\mathbb{P}_{lin}^=$, $\mathbb{NP}_{lin}^=$, \mathbb{P}_{lin}^{\leq} , and \mathbb{NP}_{lin}^{\leq} . For explicit definitions and properties of all these classes see [6,9,12,15] and especially [8] for relations to some of the major open problems in classical structural complexity theory.

Given a Boolean combination of polynomial equalities and inequalities the corresponding semi-algebraic set has at most exponentially (in n) many connected components (see [2]); checking whether two given points in the set belong to the same component can also be done in simply exponential time with respect to the number of variables (see [11]). Note that for a decision problem in NP (in any of the above models) it is easy to reduce S to another problem $\tilde{S} \in \mathbb{NP}$ such that $\tilde{S} \cap \mathbb{R}^n$ only consists of a single connected component. Just “combine” all components of $S \cap \mathbb{R}^n$ by considering the problem in \mathbb{R}^{n+1} via

$$(x_1, \dots, x_n, y) \in \tilde{S} : \Leftrightarrow (x_1, \dots, x_n) \in S \text{ or } y \neq 0.$$

However, this reduction may “destroy” the computational structure of the connected components of the original problem.

In order to obtain a meaningful notion we will combine both the number of connected components and their structure to define well-structured problems.

Definition 1. Fix one of the above variants of the BSS model. A set $S \subset \mathbb{R}^\infty$ is called well-structured (with respect to the corresponding model) iff

- (a) there exists a polynomial p such that the number of connected components of $S_n := S \cap \mathbb{R}^n$ is at most $p(n) \forall n \in \mathbb{N}$;

- (b) the function $\chi : \mathbb{R}^\infty \times S \cup S \times \mathbb{R}^\infty \rightarrow \{0, 1\}$, defined by

$$\chi(x, y) := \begin{cases} 1 & x, y \text{ belong to the same component of } S, \\ 0 & x, y \text{ belong to different components of } S, \\ 0 & x \notin S, y \in S \text{ or vice versa} \end{cases}$$

is computable in polynomial time (with respect to the specific BSS model under consideration).

Our notion is *different* from sparseness for discrete sets. This is due to the presence of condition (b). Let us therefore comment on it. Firstly, note that we do not ask for deciding whether an input belongs to $\mathbb{R}^\infty \times S \cup S \times \mathbb{R}^\infty$. If both arguments of an input do not belong to S the corresponding algorithm for χ can compute an arbitrary value—especially 1. A slightly strange flavor of this condition might be the dependence on the machine model. We will discuss this issue at the end of the paper. However, condition (b) is natural with respect to the sparseness notion over finite alphabets. There it is hidden behind the fact that equality is a polynomial time computable binary relation. Indeed, a set being sparse in the classical sense satisfies condition (b) by taking χ to be the equality relation on $(\Sigma^*)^2$ (the “components” are just single elements). One further substantiation to include condition (b) is given by the relation with non-uniform computations its presence provides. The following result holds true as well in the full BSS model as in its linear variants. It shows that well-structured sets are exactly those which can be used as oracles to capture sets having polynomial generators (where the relation between polynomial generators and non-uniform, non-deterministic computations can be established as usual, see [1]).

Proposition 1. Fix one of the above versions of the BSS model (i.e., full, linear order-free or linear ordered). Let S denote the family of all well-structured sets in the corresponding model. Then $\mathcal{NP}/\text{poly} = \bigcup_{S \in \mathcal{S}} \mathcal{NP}^S$, where \mathcal{NP} stands for $\mathbb{NP}_{\mathbb{R}}$, $\mathbb{NP}_{lin}^=$ or \mathbb{NP}_{lin}^{\leq} , respectively and oracle computations as well as non-uniform computations are defined as usual, see, for example, [19]. (In [19], non-uniformity is defined relying on families of circuits which use a finite number of constants only. In our setting, we need such cir-

circuits with a polynomially bounded number of real constants. However, the generalization of the former to the latter is straightforward and therefore omitted here).

Proof. The proof is almost straightforward from that of the corresponding result in the Turing setting [1]. Given a non-uniform, non-deterministic computation in $\mathcal{NP}/poly$ by a family of algebraic circuits $\{C_n\}$ of polynomial size (in n), these circuits can be described by polynomially many real numbers. These reals can be enlarged to vectors in \mathbb{R}^n constituting the set S_n ; each such vector represents a connected component of a single point. Clearly, $S = \bigcup_{n \geq 1} S_n$ satisfies requirement (a) of the definition of well-structuredness. It also satisfies condition (b) by taking χ as the equality relation on \mathbb{R}^∞ . The \mathcal{NP}^S computation for a given input x guesses the constants of circuit C_n (where $n = size_{\mathbb{R}}(x)$), checks by consulting the oracle S whether the guessed numbers are the correct constants and then performs the computation of C_n .

Vice versa suppose S to be a well-structured set and M to be a non-deterministic polynomial time oracle machine in the corresponding model. A polynomially sized circuit simulating M for inputs from \mathbb{R}^n can be obtained as follows: the advice the circuit uses (in form of real constants) will code a set of representatives of each connected component of S_n (without loss of generality we assume the oracle only to ask for membership in S_n , but the proof works as well for membership in any S_i with i polynomially bounded in n). Whenever the machine M enters an oracle state asking for some y to belong to S , in the circuit there will be a subroutine included for checking y to belong to the same component as one of the representatives. This subroutine uses the polynomial time algorithm originating in condition (b) of the definition of being well-structured. After a polynomial number of steps the circuit finds the correct answer and continues in simulating machine M . \square

Remark 1. In the Turing theory also

$$\mathcal{P}/poly = \bigcup_{S \in \mathcal{S}} \mathcal{P}^S$$

holds, where \mathcal{S} denotes the set of all sparse sets. The corresponding relation \supset is valid in our setting as well. However, for showing the converse one usually has to compute the advice function of a problem in

$\mathcal{P}/poly$ by means of a sparse oracle. Over the reals the latter task can never be accomplished *no matter which notion of sparseness* one defines. This is due to a simple transcendency argument when the family of constants in the circuits $\{C_n\}$ does not belong to the field extension of \mathbb{Q} by finitely many reals (the constants of the P-machine in \mathcal{P}^S).

3. Mahaney-like results for linear BSS machines

Throughout this section let us consider the linear BSS model. If no order is available then there exist problems in $\mathcal{NP}_{lin}^=$ not being reducible to a well-structured set. More explicitly, one can show

Theorem 1. *The real Knapsack decision problem is not polynomial time reducible to a well-structured set within the linear order-free BSS model.*

This result is analogous to Mahaney's original one since in the linear order-free framework the separation $\mathcal{P}_{lin}^= \neq \mathcal{NP}_{lin}^=$ is known to hold true and the Knapsack problem is one example witnessing this fact (see [12, 15]). We omit the proof here because lack of space. The main idea is to show that any reduction of the Knapsack problem to another decision problem has to be an injective map on large parts of the input. This contradicts the condition of being well-structured. The proof is similar to that showing the main result in [7].

Now we consider the linear BSS model branching on \leq . Here it is not known whether $\mathcal{P}_{lin}^{\leq} \neq \mathcal{NP}_{lin}^{\leq}$. Thus we cannot expect to show non-existence of well-structured sets, since this would prove the former separation (obviously, there are well-structured sets in \mathcal{P}_{lin}^{\leq}). Nevertheless, Mahaney's theorem holds true.

Theorem 2. *There exists an $\mathcal{NP}_{lin}^{\leq}$ -complete well-structured set if and only if $\mathcal{P}_{lin}^{\leq} = \mathcal{NP}_{lin}^{\leq}$.*

Proof. Assume a decision problem $T \subset \mathbb{R}^\infty$ belonging to $\mathcal{NP}_{lin}^{\leq}$ is given; denote by M a corresponding non-deterministic linear machine. Let S be a well-structured set such that T is polynomial time reducible by a reduction ρ to S (in the linear model with order). Without loss of generality we assume for notational simplicity that $size(\rho(x)) = size(x) \forall x \in \mathbb{R}^\infty$.

The number of connected components of $S \cap \mathbb{R}^n$ is bounded from above by a polynomial $p(n)$.

Suppose for the moment given an input dimension n we would already know a set $\{x_1, \dots, x_k\} \subset \mathbb{R}^n$ of $k \leq p(n)$ many representatives for each component of $S \cap \mathbb{R}^n$. Then for any $x \in \mathbb{R}^n$ membership in T can be decided by the application of a P_{lin}^{\leq} -algorithm to inputs of dimension n as follows: For each x_i , $1 \leq i \leq k$, compute the value of $\chi(\rho(x), x_i)$ where χ as usual denotes the function given by Definition 1 for S . Accept x if and only if at least one of the outcomes equals 1. Due to the fact that k is polynomially bounded in $size(x)$ and χ is polynomially time computable the assertion follows.

Thus it remains to be shown how to obtain $\{x_1, \dots, x_k\}$ in polynomial time. This in fact is the heart of the proof.

To do so for every input dimension n a vector $I(n) \in \{0, 1\}^*$ is constructed. This “oracle” vector codes the information we are looking for. It consists of three different parts and can be used to compute a representative set within a polynomial amount of time. Most important, the length of $I(n)$ is polynomially bounded. Thus, using an idea given in [5], the entire set of information vectors $\{I(n), n \in \mathbb{N}\}$ can be coded in a single “magic” real number s . Given n and s the information $I(n)$ can be obtained from s by a polynomial (in n) decoding process, thus providing a uniform way to access the needed information.

Before going into details let us make two more remarks. Firstly, we assume ρ to map T onto S . In general, there could exist connected components of S not containing any image point $\rho(x)$. However, in such a situation one can replace without loss of generality S by $\rho(T) =: \tilde{S} \subseteq S$. The set \tilde{S} is well-structured because it is a subset of a well structured set. Secondly, it is known from [12] that in the linear model (general) non-determinism is already captured by digital non-determinism, i.e., guessing bits instead of real numbers is sufficient to obtain the according class NP_{lin}^{\leq} . This will be useful in Step 2 below.

Step 1. Suppose the input dimension to be n , i.e., consider the reduction for $x \in \mathbb{R}^n$. Let $\rho(T \cap \mathbb{R}^n) = S_n \subset S$ and assume S_n to split into k connected components C_1, \dots, C_k . Due to S being well-structured we know k to be bounded by the value of some polynomial $p(n)$. However, using the above mentioned idea of coding a discrete set by a real number, in the linear

BSS models the census function of S is computable in polynomial time. The number k thus will represent the first part of information coded in $I(n)$. Note that $k \leq p(n)$ and therefore can be expressed by polynomially (in fact logarithmically) many bits.

Step 2. Next suppose y_1, \dots, y_k to be elements of C_1, \dots, C_k , respectively. To each y_i there corresponds a pre-image under ρ . Our aim now is to include in $I(n)$ all necessary information in order to compute points x_i such that $\rho(x_i) \in C_i$.

Let a point \tilde{x}_i in $T \cap \mathbb{R}^n$ being given such that $\rho(\tilde{x}_i) \in C_i$. To \tilde{x}_i there essentially correspond two computations; the non-deterministic machine M accepts \tilde{x}_i for at least one specific binary guess v along an accepting path γ . Here, the above mentioned equality of digital and general non-determinism is used.

For the subsequent arguments fix the guess v and note that the set V_γ of inputs branched along γ is a convex, connected subset of $T \cap \mathbb{R}^n$. Furthermore there is the computation of the reduction value $\rho(\tilde{x}_i)$. Once again a computation path δ as well as the set V_δ of inputs branched along δ can be attached to \tilde{x}_i . V_δ is convex and connected as well, and because of its polyhedral structure the same holds true for the intersection $V_\gamma \cap V_\delta$. For all connected components of S_n we collect the guesses v together with a digital coding of γ and δ . This builds up the second part of the information vector $I(n)$.

Step 3. The proof is finished if we succeed to compute a point x_i within the intersection $P := V_\gamma \cap V_\delta$. In fact, convexity of P together with affine linearity (and hence: continuity) of ρ on V_δ imply all points of P to be mapped into the same connected component of S . Since $\tilde{x}_i \in P$ this component is C_i .

The computation of such a point x_i —provided some discrete information is given—can be achieved using a technique from [16]. We therefore refer to this paper and just recall the basic ideas briefly. The set P is a polyhedron, given by polynomially many inequalities and strict inequalities:

$$P = \{x \in \mathbb{R}^n \mid A \cdot x \leq b, C \cdot x < d\}.$$

Note that all these inequalities can be explicitly obtained from the machine M , the polynomial reduction machine and the information vector $I(n)$ (its second part). Note furthermore that P is non-empty.

If P would be closed (i.e., $P = \{x \mid A \cdot x \leq b\}$), we could continue exactly as in [16]. There has to

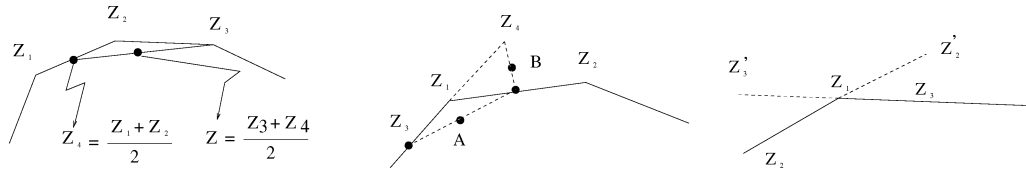


Fig. 1. (a) Number of lowest-dimensional faces is at least three. (b) Number of lowest-dimensional faces is two. (c) Number of lowest-dimensional faces is one.

be a point in P on a face of lowest dimension, and all such points are solutions of a specific subsystem $\tilde{A} \cdot x = \tilde{b}$ of linear equations; these equations arise by replacing the “ \leq ” in some of the original inequalities by an “ $=$ ”. The information about which inequalities are active in order to obtain $\tilde{A} \cdot x = \tilde{b}$ can be coded by a vector of polynomially many bits. However, the presence of strict inequalities enforces a little more subtle reasoning.

We first proceed as above, this time for the closure \overline{P} of P which is given by

$$\overline{P} = \{x \in \mathbb{R}^n \mid A \cdot x \leq b, C \cdot x \leq d\}.$$

Now the foregoing depends on the number of lowest-dimensional faces \overline{P} has.

In case this number is at least three compute in a similar way as described above three points $z_1, z_2,$ and z_3 on three different such faces. Now

$$z = \frac{1}{4} \cdot (z_1 + z_2) + \frac{1}{2} \cdot z_3$$

belongs to P .

If there are only two faces of lowest dimension first compute one point on each of them, say z_1 and z_2 . Since z_1 is laying on a face of lowest dimension it has to belong to the intersection of at least two faces of higher dimension (if the two faces are parallel a point in P is given by $\frac{1}{2} \cdot (z_1 + z_2)$). At least one of these faces does not contain z_2 . Compute any arbitrary point z_3 on that face (possibly outside \overline{P}). Afterwards compute

$$z_4 := 2 \cdot z_1 - z_3$$

which is also lying on that face. We obtain a point in P among one of the points

$$A = \frac{1}{4} \cdot (z_1 + z_2) + \frac{1}{2} \cdot z_3 \quad \text{and}$$

$$B = \frac{1}{4} \cdot (z_1 + z_2) + \frac{1}{2} \cdot z_4.$$

Finally, if there is only one face of lowest dimension and z_1 denotes a point on it, the face must lay in the intersection of two other faces of higher dimension. On each of them we compute an arbitrary point z_1 and z_2 together with its reflection z'_1 and z'_2 with respect to z_1 . One of the lines combining the four corresponding points has its middle-point in P (see Fig. 1).

All the information which allows to obtain the correct equations of the important faces as well as a point on it can be coded in a polynomially sized vector of bits. This vector (again collected for all components) constitutes the third part of $I(n)$.

The proof is finished. \square

4. Concluding remarks; open problems

Beside Mahaney’s theorem there are numerous further results concerning the properties of sparse sets. In particular there is a strong relationship between sparseness and the collapse of higher time complexity classes in the Turing model. More explicitly, Hartmanis [10] has shown that $\text{NEXP} \neq \text{EXP}$ if and only if there exists a sparse set in $\text{NP} \setminus P$.

In the linear ordered BSS model one can show

Theorem 3. *If $\text{NEXP}_{lin}^{\leq} = \text{EXP}_{lin}^{\leq}$ every well-structured set in NP_{lin}^{\leq} is in P_{lin}^{\leq} .*

The proof once more mainly relies on the technique of coding a point in a polyhedron by a vector of digits used in Theorem 2. The reverse implication remains open.

One important feature of Theorem 2 is its different proof in comparison with the one of Mahaney’s original theorem. Mahaney’s proof relies on the fact that there exists an NP-complete problem being self-reducible; it thus only works for self-reducible prob-

lems. Contrary, the proof of Theorem 2 shows that any problem being reducible to a well-structured set in the ordered linear BSS model is polynomial time decidable. It does not depend on a specific structure of complete problems in NP_{lin}^{\leq} (the existence of complete problems was established in [9]).

Even in the full BSS model Mahaney's proof, applied to our notion and to a self-reducible problem will work. However, it is not at all known whether there exist such problems being $\text{NP}_{\mathbb{R}}$ -complete. We thus consider it to be important presenting a proof without a self-reducibility assumptions on the structure of the problems.

As is already obvious from the above proof the coding power of real numbers turns out to be extremely helpful. With respect to well-structured sets it provides another effect which is unknown in the discrete setting. There it is just known that a sparse set in NP belongs to the second level L_2 of Schöning's low hierarchy (see [20,1]). It is not known whether such a set already belongs to the first level $L_1 = \text{NP} \cap \text{co-NP}$. In real models the situation is different. Since any function from \mathbb{N} to \mathbb{N} is BSS computable this especially holds true for the census function of the number of connected components. Thus if any set $S \subset \mathbb{R}^{\infty}$ is well-structured and belongs to the complexity class $\text{NP}_{\mathbb{R}}$ (or to the NP class in any other of the models considered above), its complement $\mathbb{R}^{\infty} \setminus S$ belongs to $\text{NP}_{\mathbb{R}}$ as well. This can be seen as follows: For an input $x \in \mathbb{R}^n$ first compute the census $p(n)$ of $S \cap \mathbb{R}^n$. Due to the well-structuredness of S the census p is a polynomial and thus the entire sequence of integers $p(1), p(2), p(3), \dots$ can be coded in a single real number s such that each value $p(n)$ can be decoded from s in polynomial time with respect to n (see [5]). Next, guess $p(n)$ many reals $x_1, \dots, x_{p(n)}$ in $S \in \text{NP}_{lin}^{\leq}$ such that no two of them belong to the same component. The latter can be done in polynomial time according to S being well structured. Finally, accept x if and only if $\chi(x, x_i) = 0 \forall i \in \{1, \dots, p(n)\}$. Then x is accepted iff $x \in \mathbb{R}^{\infty} \setminus S$.

Therefore, the assumption of the existence of a $\text{NP}_{\mathbb{R}}$ -complete well-structured set will already imply $\text{NP}_{\mathbb{R}} = \text{co-NP}_{\mathbb{R}}$. However, there are of course undecidable well-structured sets and Proposition 1 proves our notion not being too strong.

Finally let us consider again condition (b) in Definition 1.

A purely topological notion of sparseness over the reals was introduced in [7]. There, a Mahaney-like result was shown for the order-free linear case. It seems to be not at all clear how to use such a purely topological property for obtaining Mahaney's theorem in models other than order-free linear machines, if there is not any additional "computable" information available about the "sparse" object. As we have seen in the classical setting the latter intrinsically is given in a hidden way. For the real models we believe that including such an information could help to provide meaningful concepts of sparseness. In the case of measuring the dimension of a set [7] such an additional knowledge could address the structure of the semi-algebraic maps which relate a given semi-algebraic set of dimension d to \mathbb{R}^d . In [17] a specialization of the sparseness notion used in [7] and including such an additional computability property is investigated to relate exponential time classes in the BSS model with the complexity of the Boolean Satisfiability problem within real models.

Referring to the approach towards a sparseness-like notion followed in the present paper condition (b) could play the role of such an information.

It remains an open question to study well-structured sets in the full BSS model. This holds true as well with respect to Mahaney's theorem as to the relations with exponential time classes. It remains open as well to study whether there are closer relationships between our notion and the classical one beside the fact that they imply similar properties for the corresponding computational models.

Acknowledgment

This work was done while the second author was visiting the Universidad de Chile. The stay was made possible as part of the FONDAF program in Applied Mathematics. We gratefully acknowledge this support.

References

- [1] J.L. Balcázar, J. Diaz, J. Gabarró, Structural Complexity I, II, EATCS Monographs of Theoretical Computer Science, Vol. 11, Vol. 12, Springer, Berlin, 1988, 1990.
- [2] R. Benedetti, J.J. Risler, Real Algebraic and Semi-Algebraic Sets, Hermann, Paris, 1990.

- [3] L. Blum, F. Cucker, M. Shub, S. Smale, Complexity and real computation: A manifesto, *Internat. J. Bifurcation Chaos* 6 (1996) 3–26.
- [4] L. Blum, F. Cucker, M. Shub, S. Smale, *Complexity and Real Computation*, Springer, Berlin, 1997.
- [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] F. Cucker, P. Koiran, Computing over the reals with addition and order. Higher complexity classes, *J. Complexity* 11 (3) (1995) 358–376.
- [7] F. Cucker, P. Koiran, M. Matamala, Complexity and Dimension, *Inform. Process. Lett.* 62 (1997) 209–212.
- [8] H. Fournier, P. Koiran, Are lower bounds easier over the reals?, in: *Proc. Symposium on Theory of Computing*, 1998.
- [9] C. Gassner, On NP-completeness for linear machines, *J. Complexity* 13 (2) (1997) 259–271.
- [10] J. Hartmanis, On sparse sets in NP-P, *Inform. Process. Lett.* 16 (1983) 55–60.
- [11] J. Heintz, M.F. Roy, P. Solernò, Single exponential path finding in semi-algebraic sets II: The general case, in: C.L. Bajaj (Ed.), *Algebraic Geometry and its Applications*, Springer, Berlin, 1993, pp. 467–481.
- [12] P. Koiran, Computing over the reals with addition and order, *Theoret. Comput. Sci.* 133 (1994) 35–47.
- [13] W. Maass, Bound for the computational power and learning complexity of analog neural nets, in: *Proc. 25th Symposium on the Theory of Computing*, 1995, pp. 335–344.
- [14] S.R. Mahaney, Sparse complete sets for NP: Solution of a conjecture by Berman and Hartmanis, *J. Computer System Sci.* 25 (1982) 130–143.
- [15] K. Meer, A note on a $P \neq NP$ -result for a restricted class of real machines, *J. Complexity* 8 (1991) 451–453.
- [16] K. Meer, On the complexity of quadratic programming in real number models of computations, *Theoret. Comput. Sci.* 133 (1994) 85–94.
- [17] K. Meer, On the relations between discrete and continuous complexity theory, *Math. Logic Quarterly* 41 (1995) 281–286.
- [18] K. Meer, C. Michaux, A survey on real structural complexity theory, *Bull. Belgian Math. Soc. Simon Stevin* 4 (1997) 113–148.
- [19] B. Poizat, *Les Petits Cailloux*, Aléas, Lyon, 1995.
- [20] U. Schöning, A low and an high hierarchy within NP, *J. Comput. System Sci.* 27 (1983) 14–28.